COMPAQ® ENHANCED COLOR GRAPHICS BOARD
COMPAQ COLOR MONITOR

TECHNICAL REFERENCE GUIDE

COMPAQ®

The COMPAQ® Enhanced Color Graphics Board

TECHNICAL REFERENCE GUIDE

NOTICE

The information in this guide is subject to change without notice. COMPAQ® Computer Corporation shall not be liable for technical or editorial omissions made herein; nor for incidental or consequential damages resulting from the furnishing, performance, or use of this material.

COMPAQ Computer Corporation reserves the right to make improvements to products without incurring any obligation to incorporate such improvements into products previously sold.

This guide contains information protected by copyright. No part of this guide may be photocopied or reproduced in any form without prior written consent from COMPAQ Computer Corporation.

Copyright © 1986 by COMPAQ Computer Corporation.
All rights are reserved.

COMPAQ®, COMPAQ Portable, COMPAQ PLUS® Personal Computer, COMPAQ PORTABLE 286® Personal Computer, and COMPAQ PORTABLE II® Personal Computer are Registered Trademarks of COMPAQ Computer Corporation. COMPAQ DESKPRO 386™ Personal Computer is a Trademark of COMPAQ Computer Corporation.

The COMPAQ Enhanced Color Graphics Board is covered by one or more of the following U.S. Patents:

4,574,279
4,197,590 re 31,200

The software described in this guide is furnished under a license agreement. The software may be used or copied only in accordance with the terms of the agreement.

MS DOS® is a Registered Trademark of Microsoft Corporation.

Copyright © 1981, 1982, 1983, 1984, 1985 by Microsoft Corporation.

IBM PC AT® is a Registered Trademark of International Business Machines, Inc.

Intel® is a Registered Trademark of Intel.

Microsoft® is a Registered Trademark of Microsoft Corporation.

COMPAQ Enhanced Color Graphics Board
Technical Reference Guide
First Edition (December 1986)
Assembly #106450-001
Book # 106451-001
Binder # 106452-001
Option Kit # 106465-001
Owner Information Card # 108276-002

COMPAQ Computer Corporation

COMPAQ COMPUTER CORPORATION COMPAQ ENHANCED COLOR GRAPHICS BOARD ONE-YEAR LIMITED WARRANTY

Definitions

This one-year limited Warranty applies to the COMPAQ Color Monitor and COMPAQ Enhanced Color Graphics Board. The COMPAQ Color Monitor and the COMPAQ Enhanced Color Graphics Board are the "COMPAQ Products" referred to in this Warranty.

You are the "Purchaser" if you are the original purchaser of the COMPAQ Product from either Compaq Computer Corporation ("Compaq") or an Authorized COMPAQ® Computer Dealer. This Warranty only applies to original Purchasers, and is not transferable to anyone who purchases the COMPAQ Product from you.

The "Date of Purchase" is the date you originally purchased the COMPAQ Product, and is shown on the original Authorized COMPAQ Computer Dealer or Compaq proof of purchase. Your sales receipt, showing the Date of Purchase, the COMPAQ Product(s) purchased, and the name of the Authorized COMPAQ Computer Dealer that sold you the COMPAQ Product(s), serves as your proof of the Date of Purchase.

Warranty

Compaq warrants to you, the Purchaser, that the COMPAQ Color Monitor and/or the COMPAQ Enhanced Color Graphics Board you have purchased from an Authorized COMPAQ Computer Dealer or from Compaq is free from defects in materials and workmanship for a period of one year from the Date of Purchase.

Remedies

During the term of this Warranty, at no additional charge to the Purchaser, Compaq will repair or replace any defective parts of the COMPAQ Color Monitor and/or COMPAQ Enhanced Color Graphics Board purchased under this limited Warranty, with new or, at the option of Compaq, refurbished parts. If Compaq exercises its option to replace the COMPAQ Color Monitor and/or the COMPAQ Enhanced Color Graphics Board, the replacement COMPAQ Product(s) shall be covered under this warranty for the balance of the one-year period from the date of original purchase or for at least ninety (90) days whichever is longer. To obtain this repair or replacement service, the Purchaser must return the COMPAQ product to an Authorized COMPAQ Computer Dealer or to Compaq. The Purchaser also may return the COMPAQ Product to a participating Authorized COMPAQ Computer Warranty Service Provider, at the Service Provider's option. The Purchaser must prepay any shipping charges. In addition, the Purchaser is responsible for insuring any product so returned and assumes the risk of loss during shipping. All parts and products replaced under this Warranty become the property of Compaq.

Warranty Claim Requirements

To claim service pursuant to this Warranty you, the Purchaser, must:

1. Provide any Authorized COMPAQ Computer Dealer, Compaq, or a participating Authorized COMPAQ Computer Warranty Service Provider with proof of the Date of Purchase from an Authorized COMPAQ Computer Dealer or Compaq.

2. Return the COMPAQ Product to an Authorized COMPAQ Computer Dealer, Compaq, or at the Service Provider's option, a participating Authorized COMPAQ Computer Warranty Service Provider. The Purchaser must prepay any shipping charges. In addition, the Purchaser is responsible for insuring any product shipped for return, because under this Warranty, the Purchaser assumes the risk of loss during shipping.

3. The Purchaser may request information on how to obtain Warranty service by contacting any Authorized COMPAQ Computer Dealer. You may call 1-800-231-0900, toll-free, from anywhere in the 48 contiguous United States and Hawaii for the location of the Authorized COMPAQ Computer Dealer nearest you; or

4. The Purchaser may also request information on how to obtain Warranty service by writing to the Customer Relations Department at Compaq Computer Corporation, 20555 FM 149, Houston, Texas 77070.

Disclaimers

THIS LIMITED WARRANTY DOES NOT EXTEND TO ANY PRODUCTS NOT PURCHASED FROM COMPAQ OR AN AUTHORIZED COMPAQ COMPUTER DEALER. THIS LIMITED WARRANTY ALSO DOES NOT EXTEND TO ANY PRODUCTS WHICH HAVE BEEN DAMAGED OR RENDERED DEFECTIVE (a) AS A RESULT OF ACCIDENT, MISUSE, OR ABUSE; (b) BY THE USE OF PARTS NOT MANUFACTURED OR SOLD BY COMPAQ; (c) BY MODIFICATION; OR (d) AS A RESULT OF SERVICE BY ANYONE OTHER THAN COMPAQ, AN AUTHORIZED COMPAQ COMPUTER DEALER, OR AN AUTHORIZED COMPAQ COMPUTER WARRANTY SERVICE PROVIDER.

EXCEPT AS EXPRESSLY SET FORTH ABOVE, COMPAQ MAKES NO OTHER WARRANTIES, EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, ANY IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE, AND COMPAQ EXPRESSLY DISCLAIMS ALL WARRANTIES NOT STATED HEREIN. IN THE EVENT THE PRODUCT IS NOT FREE FROM DEFECTS AS WARRANTED ABOVE, YOUR SOLE REMEDY AS THE PURCHASER SHALL BE REPAIR OR REPLACEMENT AS PROVIDED ABOVE. UNDER NO CIRCUMSTANCES WILL COMPAQ BE LIABLE TO THE PURCHASER, OR TO ANY USER, FOR ANY DAMAGES, INCLUDING ANY INCIDENTAL OR CONSEQUENTIAL DAMAGES, EXPENSES, LOST PROFITS, LOST SAVINGS, OR OTHER DAMAGES ARISING OUT OF THE USE OF OR INABILITY TO USE THE COMPAQ PRODUCT.

ANY IMPLIED WARRANTIES ARE LIMITED TO THE TERM OF THIS EXPRESS LIMITED WARRANTY.

SOME STATES DO NOT ALLOW THE EXCLUSION OR LIMITATION OF INCIDENTAL OR CONSEQUENTIAL DAMAGES FOR CONSUMER PRODUCTS, AND SOME STATES DO NOT ALLOW LIMITATIONS ON HOW LONG AN IMPLIED WARRANTY LASTS, SO THE ABOVE LIMITATIONS OR EXCLUSIONS MAY NOT APPLY TO YOU.

THIS WARRANTY GIVES YOU SPECIFIC LEGAL RIGHTS, AND YOU MAY ALSO HAVE OTHER RIGHTS WHICH VARY FROM STATE TO STATE.

PREFACE

This guide provides hardware and firmware (ROM) information for the COMPAQ® Enhanced Color Graphics (CECG) Board for:

- Developers who want to know more about the hardware and the general theories of how the COMPAQ Enhanced Color Graphics Board works.
- Technicians or engineers who need technical information to design accessories for the COMPAQ Enhanced Color Graphics Board.
- Programmers who need to know about the hardware (programmable devices) and firmware for programming purposes.
- Individuals or companies who want to know how the COMPAQ Enhanced Color Graphics Board operates and how to access its features.

This document assumes an understanding of the Intel 8088, 8086, 80286, and 80386 microprocessors and of personal computers based on those microprocessors. Familiarity with one of the following COMPAQ manuals is helpful to understanding this document.

- 80286-Based Products Technical Reference Guide (part no. 102786-001)
- COMPAQ PORTABLE II® Technical Reference Guide (part no. 104456-001)
- COMPAQ DESKPRO 386™ Technical Reference Guide (part no. 108037-001)

For information about Intel publications, write or call:

Intel Literature Sales
P. O. Box 58130
Santa Clara, CA 95052-8130
800-548-4725
Source: CQ

The Intel documents apply only to Intel Corporation manufactured parts and parts from Intel licensees.

For more information about COMPAQ publications, write:

COMPAQ Computer Corporation
20555 FM 149
Houston, Texas 77070

HOW THIS DOCUMENT IS ORGANIZED

This book is divided into seven chapters and three appendixes.

Chapter 1 gives information about the COMPAQ Enhanced Color Graphics Board features. This chapter is a system overview and should be read first. Chapters 2 through 7 give information about the theory of operation, specifications, register programming, BIOS, configuration, connectors, jumpers, switches, and schematics.

The appendixes contain information on the COMPAQ Color Monitor, programming examples, and monitor switching via software.

An index is provided for your convenience.

NOTATIONAL CONVENTIONS

Values

I/O addresses and other values are in hexadecimal notation when shown with the lowercase letter "h" as a suffix. Memory addresses are always in hexadecimal notation and are expressed as SSSS:OOOO, where SSSS is a 16-bit segment and OOOO is a 16-bit offset.

Ranges

Ranges or limits on a parameter are shown as a pair of values separated by two dots (..). For example, 4..0 includes numbers 4, 0, and every number in between (3, 2, and 1).

Signal Labels

Signals are labeled, for example, as A0, A1, A15, and so on. Signal names are in uppercase letters. Signals that are negative true, or active when low, are indicated in the text by a dash (-) suffix; in the schematics, negative true signals are followed by an asterisk (*).

Bit values are labeled as bit 7, bit 6, bit 5, and so on.

Register Notation and Usage

The standard Intel naming conventions are used for the CPU registers. AX, BX, CX, and DX are the names of the general registers when used as word-length registers (16-bit). AH, AL, BH, BL, CH, CL, DH, and DL are the names for the general registers when they are used as byte-length registers (8-bit). When addresses are handled, BX usually contains the offset. However, SI (source index) or BP (base pointer) may also be used with the ES (extra segment) register.

Register-set boxes that are shaded are ignored on input or unchanged for output. An exception is that the contents of AX are not guaranteed to be preserved across all calls. Always reload the function code in AH and the parameter in AL (if any) to repeat a call.

Bit Notation

Bit fields within a byte or word are shown as a range of decimal numbers separated by two dots enclosed in angle brackets <..>. For example, reference to the four most-significant bits in a word is made as <15..12>. The higher number, representing the most-significant bit, is on the left.

Common Abbreviations

The following acronyms and abbreviations are used throughout this guide:

Acronym	Meaning
AC	alternating current
BIOS	bidirectional input/output system
CGA	color graphics adapter
CRT	cathode ray tube
D/A	digital-to-analog
DC	direct current
dword	double word (32-bit)
EGA	enhanced graphics adapter
HW	hardware
INT	interrupt
lsb	least-significant bit
MDA	monochrome display adapter
MSB	most-significant byte
PTR	pointer
RAM	random-access memory
ROM	read-only memory
SW	software
VDU	video display unit

Units of Measure

Symbol	Meaning
A	ampere
^{o}C	degrees Celsius
^{o}F	degrees Fahrenheit
Hz	hertz
KB	kilobyte
kHz	kilohertz
mb	megabit
MB	megabyte
ms	millisecond
MHz	megahertz
uF	microfarad
n	variable parameter/value
us	microsecond
V	volt
W	watt

ABOUT COMPATIBLES

Throughout this document reference is made to COMPAQ products or a "compatible." Compatible refers to alternative products from third-party suppliers that, given adherence to the specified standard, should operate as stated herein. COMPAQ does not guarantee that all third-party products claiming to be compatible with the specified standard will be fully functional.

Examples of compatible digital-type monitors and adapters are:

CGA Compatible = A product such as the IBM® Color Graphics Adapter.

Enhanced Color Monitor (RrGgBb) Compatible = A monitor that has the capability of displaying 64 colors, such as the IBM® Model 5154 Monitor.

MDA Compatible = A product such as the IBM® Monochrome Display Adapter.

Monochrome Compatible = A monitor that has the capability of displaying three levels of intensity, such as the IBM® Model 5151 Monitor.

RGB Compatible = A monitor that has the capability of displaying 8 colors.

RGBI Compatible = A color monitor that has the capability of displaying 16 colors, such as the IBM® Model 5153 Monitor.

CONTENTS

PREFACE

HOW THIS DOCUMENT IS ORGANIZED

NOTATIONAL CONVENTIONS

ABOUT COMPATIBLES

CHAPTER 1
OVERVIEW

CHAPTER 2
THEORY OF OPERATION

2.1 OVERVIEW 2-1

2.2 BLOCK DIAGRAMS AND THEIR DESCRIPTIONS 2-12

2.3 POWER-ON SELF-TEST CODES 2-57

CHAPTER 3
SPECIFICATIONS

3.1 HARDWARE 3-1

3.2 SOFTWARE 3-1

CHAPTER 4
REGISTER PROGRAMMING

4.1 I/O ADDRESSES 4-1

4.2 CONTROL AND STATUS REGISTERS 4-2

4.3 SEQUENCER REGISTERS 4-9

4.4 GRAPHICS CONTROLLER REGISTERS 4-13

4.5 CRT CONTROLLER REGISTERS 4-21

4.6 ATTRIBUTE CONTROLLER REGISTERS 4-38

4.7 MODE-SPECIFIC REGISTER VALUES 4-43

CHAPTER 5
BIOS

5.1 INTRODUCTION TO VIDEO BIOS 5-1

5.2 VIDEO BIOS MODES SUPPORTED 5-2

5.3 DEFAULT COLOR PALETTE 5-4

5.4 INITIALIZATION 5-6

5.5 POWER-ON SELF-TEST 5-6

5.6 INTERRUPTS 5-6

CHAPTER 6
CONFIGURING THE BOARD

6.1 VIDEO MEMORY SPACING AND I/O ADDRESSING 6-1

6.2 SWITCHES AND JUMPERS 6-2

6.3 VIDEO CONFIGURATIONS FOR PORTABLE COMPUTERS 6-8

6.4 VIDEO CONFIGURATIONS FOR DESKTOP COMPUTERS 6-9

6.5 CONNECTORS 6-10

6.6 OUTPUT SIGNAL WAVEFORMS 6-17

CHAPTER 7
SCHEMATICS

APPENDIX A
COMPAQ COLOR MONITOR

A.1 Introduction A-1
A.2 Specifications A-3
A.3 Installation Considerations A-3
A.4 Self-Test A-6
A.5 Connections A-7
A.6 Signal Timing Waveforms A-8

APPENDIX B
PROGRAMMING EXAMPLES

B.1 CLEARING THE SCREEN B-1
B.2 LOADING THE CHARACTER FONT B-7
B.3 SHOWING A SPLIT SCREEN B-19
B.4 LOGICAL WINDOWING B-26
B.5 SMOOTH SCROLLING AND PANNING B-35
B.6 USING THE VERTICAL BLANKING INTERRUPT B-62
B.7 PROGRAMMING THE COLOR PALETTE B-77
B.8 MODIFYING VIDEO PARAMETERS B-82
B.9 MAKING SPECIAL FONTS RESIDENT B-89
B.10 SWITCHING BETWEEN COLOR TEXT AND MONOCHROME TEXT DISPLAY MODES B-98
B.11 MISCELLANEOUS ROUTINES B-109

APPENDIX C
MONITOR SWITCHING VIA SOFTWARE

INDEX

LIST OF FIGURES

FIGURE

2-1. Video RAM Planes 2-2
2-2. Block Diagram of the Master Clock Selection 2-7
2-3. Block Diagram of the Text Mode (BIOS Modes 0, 1, 2, 3, 7, 0*, 1*, 2*, and 3*) 2-15
2-4. Definition of the Attribute Byte 2-16
2-5. Format of the Font Address in Plane 2 2-18
2-6. Matrix for Character "A" in the 8 x 14 Font 2-20
2-7. Color Selection in the Monitor 2-22
2-8. Comparison of Linear Addressing with Interleaved Addressing 2-24
2-9. Relationship Between Bit Pairs and Pixels in the Compatible Color Graphics Mode (BIOS Modes 4 and 5) 2-24
2-10. Block Diagram of the Compatible Color Graphics Mode (BIOS Modes 4, 5, and 6) 2-26
2-11. Data Flow of BIOS Modes 4 and 5 2-27
2-12. Programmable Logic Device for the COMPAQ Color Monitor 2-30
2-13. Programmable Logic Device for the COMPAQ Dual-Mode Monitor 2-31
2-14. Data Flow from BIOS Mode 6 2-33
2-15. Block Diagram of the Enhanced Graphics Mode (BIOS Modes D, F, and 10) 2-35
2-16. Data Rotate Register 2-38
2-17. Data Rotation 2-38
2-18. Enable Bit Set/Reset Register 2-39
2-19. Write Mode Selector 2-40
2-20. Data Rotate Register 2-41
2-21. Bit Mask Register 2-42
2-22. Bit Mask Logic Output 2-43
2-23. Read Mode 0 Data Path 2-44
2-24. Data Display Cycle 2-46
2-25. Color Comparison 2-48
2-26. Results of Color Comparison 2-49
2-27. Lightpen Functional Block Diagram 2-56
2-28. Video Memory Chip Layout and Location Codes 2-62
5-1. Illustration of the Feature Bits Register 5-36
5-2. Character Cell in 8 x 8 Character Font 5-44
5-3. Layout of the Graphics 8 x 8-Dot Table for the Upper-128-Character Set 5-45
5-4. Character Cell in 8 x 14 Character Font 5-47
5-5. Character Cell in the 9 x 14 Character Font 5-47
5-6. Layout of the Graphics 8 x 14-Dot Table for the Entire 256-Character Set 5-48

5-7. Illustration of the Bit Assignments of the Word at Address 0040:0010 5-50
5-8. Illustration of the Bit Assignments of the Byte at Address 0040:0087 5-50
5-9. Illustration of the Bit Assignments of the Byte at Address 0040:0088 5-51
5-10. Illustration of the Bit Assignment of the Byte at Address 0040:0100 5-54
6-1. Switch Bank and Jumper Locations 6-2
6-2. SW2 Configuration Settings 6-6
6-3. Connector Locations 6-10
6-4. J1 RCA Connector 6-10
6-5. J2 RCA Connector 6-10
6-6. J4, Special Features Connector 6-12
6-7. J401, Internal Monitor Connector 6-13
6-8. J404 Connector for the COMPAQ Dual-Mode Monitor or a Compatible RGBI Color Monitor 6-15
6-9. J404 Connector for the COMPAQ Color Monitor or a Compatible Enhanced Color Monitor 6-16
6-10. J405, Lightpen Interface Connector 6-16
6-11. Horizontal Timing 6-17
6-12. Vertical Timing 6-17
6-13. Digital-to-Analog Conversion Characteristics for the COMPAQ Dual-Mode Monitor Video Display 6-18
7-1. COMPAQ Enhanced Color Graphics Board Schematics 7-1
A-1. COMPAQ Color Monitor A-1
A-2. COMPAQ Color Monitor Functional Block Diagram A-2
A-3. COMPAQ Color Monitor Controls A-6
A-4. COMPAQ Color Monitor Signal Connector A-7
A-5. COMPAQ Color Monitor Horizontal Timing A-8
A-6. COMPAQ Color Monitor Vertical Timing A-8
B-1. Example of a Split Screen B-19
B-2. Relationship Between a Logical Window and Its Borders B-26

LIST OF TABLES

TABLE

1-1. Monitors and Modes Supported by the COMPAQ Enhanced Color Graphics Board 1-1
2-1. BIOS Modes and Their Corresponding Video Outputs 2-6
2-2. Master Clock Frequencies 2-7
2-3. Operating Modes and Their Corresponding BIOS Modes and Formats 2-8
2-4. Text BIOS Modes, Their Descriptions, and Monitors Supported 2-9
2-5. Compatible Color Graphics BIOS Modes, Their Descriptions, and Monitors Supported 2-10
2-6. Enhanced Graphics BIOS Modes, Their Descriptions, and Monitors Supported 2-11
2-7. Color Palette Values with Registers Programmed for Mode 3* (80 x 25 Text) 2-21
2-8. Colors Selected by Attribute Bits 2-25
2-9. Conversion of RrGgBb Colors to RGBI Colors 2-52
2-10. Conversion of VI Signals to RGBI Intensities 2-54
2-11. BIOS ROM Power-On Self-Test Codes and Their Explanations 2-57
3-1. Board Connectors 3-1
3-2. Control and Status Registers 3-1
3-3. Sequencer Registers 3-2
3-4. Graphics Controller Registers 3-2
3-5. CRT Controller Registers 3-2
3-6. Attribute Controller Registers 3-3
3-7. BIOS Mode Operating Characteristics 3-4
3-8. Listing of Specific I/O Addresses 3-5
3-9. Video BIOS Calls 3-5
4-1. I/O Addresses for Registers 4-1
4-2. Control and Status Registers 4-2
4-3. Master Clock Frequencies 4-6
4-4. Sequencer Registers 4-9
4-5. Graphics Controller Registers 4-13
4-6. CRT Controller Registers 4-22
4-7. CRT Controller Indices 4-23
4-8. Typical Cursor Register Programming 4-29
4-9. Attribute Controller Registers 4-39
4-10. Attribute Controller Registers 4-44
5-1. BIOS Modes and Supported Monitors 5-2
5-2. Default Colors for a 4-Color Palette 5-4
5-3. Default Colors for a 16-Color Palette 5-5
5-4. Default Colors for a 2-Color Palette 5-5
5-5. Default Intensities for a 3-Intensity Monochrome Text Display 5-5

5-6. Default Intensities for a 3-Intensity/Blinking Monochrome Graphics Display 5-5
5-7. Video Interrupts 5-6
5-8. Memory Locations Used by Video Interrupts 5-7
5-9. Vertical Screen Blanking Time 5-8
5-10. INT 10h Video BIOS Calls 5-9
5-11. CL Register Codes and Their Corresponding Video Configurations 5-37
5-12. Video BIOS RAM Locations 5-49
5-13. Listing of Parameter Table Pointers 5-52
5-14. Video Parameter Table (1472 Bytes Total) 5-52
5-15. Dynamic Save Area Table 5-53
5-16. Text Mode Auxiliary Character Generator Table 5-53
5-17. Graphics Mode Auxiliary Character Generator Table 5-54
6-1. Operating Address Ranges for EGA-Class Controllers According to Emulation 6-2
6-2. SW1 Settings with Only the COMPAQ Enhanced Color Graphics Board in the System 6-3
6-3. SW1 Settings for Two Video Display Controllers in the Same System with the COMPAQ Enhanced Color Graphics Board as the Primary Controller 6-4
6-4. SW1 Settings for Two Video Display Controllers in the Same System with the COMPAQ Enhanced Color Graphics Board as the Secondary Controller 6-5
6-5. Jumper P1 Settings 6-7
6-6. Jumper JP3 Settings 6-7
6-7. SW1 and SW2 Settings for All Possible Video Configurations for COMPAQ Portable Computers 6-8
6-8. SW1 and SW2 Settings for All Possible Video Configurations for COMPAQ Desktop Computers 6-9
6-9. J1 and J2 Signals 6-10
6-10. J4 Connector Signals 6-11
6-11. J401 Connector Signals 6-13
6-12. Comparison of Signal Functions 6-14
6-13. J404 Connector Signals for COMPAQ Dual-Mode Monitor or a Compatible RGBI Color Monitor 6-14
6-14. J404 Connector Signals for the COMPAQ Color Monitor or a Compatible Enhanced Color Monitor 6-15
6-15. J405, Lightpen Interface Signals 6-16
A-1. COMPAQ Color Monitor Specifications A-3
A-2. Monitor Connector Signal Descriptions A-7
B-1. Vertical Blanking Interval B-62
C-1. Video Display Responses to the MODE Command with the COMPAQ Enhanced Color Graphics Board as the Only Video Display Controller Installed (COMPAQ Desktop Computers) C-2
C-2. Video Display Responses to the MODE Command with the COMPAQ Enhanced Color Graphics Board as the Primary Video Display Controller and the COMPAQ Video Display Controller (VDU) as the Secondary One (COMPAQ Desktop Computers) C-3
C-3. Video Display Responses to the MODE Command with the COMPAQ Enhanced Color Graphics Board as the Primary Video Display Controller and a Monochrome Video Adapter as the Secondary One (COMPAQ Desktop Computers) C-4
C-4. Responses to CTRL + ALT + < and CTRL + ALT + > Commands (COMPAQ Portable Computers) C-5
C-5. Responses to CTRL + ALT + < and CTRL + ALT + > Commands (COMPAQ Desktop Computers) C-9

TABLE OF CONTENTS

CHAPTER 1
OVERVIEW

Chapter 1
OVERVIEW

The COMPAQ® Enhanced Color Graphics Board is an enhanced graphics adapter (EGA)-compatible video display controller board. It is a versatile color board that also provides the features of color graphics adapters (CGA), monochrome display adapters (MDA), and the COMPAQ Video Display Unit (VDU) Controller board. The board generates high-resolution text and graphics in as many as 16 colors simultaneously from a palette of 64 colors. It can be used as the only video display controller board in your system or in conjunction with another video display controller board installed in the same system.

Features of the COMPAQ Enhanced Color Graphics Board include a:

- Graphics resolution of 640 x 350, which allows a full screen of 8 x 14 dot characters in color.
- Palette of 64 colors, which provides a wide range of software-accessible color selections.
- Standard 256 kilobytes (KB) of video RAM, allowing full 16-color displays within the 64-color palette.
- Internal monitor connector interface for use in COMPAQ Portable Computers, designed specifically to allow the connection of an internal 9-inch COMPAQ Dual-Mode Monitor in the COMPAQ Portable , COMPAQ PLUS®, COMPAQ PORTABLE 286®, and COMPAQ PORTABLE II® Personal Computers. This interface is a COMPAQ-unique feature.
- Lightpen interface, which accommodates applications that use an industry-standard lightpen device.

The COMPAQ Enhanced Color Graphics Board supports three modes of operation: enhanced color, color, and monochrome. Each mode supports certain monitors, which are listed in Table 1-1.

Table 1-1. Monitors and Modes Supported by the COMPAQ Enhanced Color Graphics Board

	Modes of Operation		
Monitor Type	Enhanced Color	Color	Monochrome
COMPAQ Color Monitor	X	X	-
COMPAQ Dual-Mode Monitors:			
COMPAQ DESKPRO® 12" (green or amber)	X	X	X
Portable Internal 9"	X	X	X
Compatible RGBI monitor (16 possible colors)	-	X	-
Compatible RGB monitor (8 possible colors)	-	X	-
Compatible Enhanced Color Monitor	X	X	-

Legend: X = Supported
- = Not supported

TABLE OF CONTENTS

CHAPTER 2
THEORY OF OPERATION

2.1 OVERVIEW 2-1
  Features 2-1
    Color Palette 2-2
    Video RAM Planes 2-2
    Modifiable Text Fonts 2-2
    Logical Windowing 2-3
    Smooth Scrolling and Panning 2-3
    Split Screen 2-4
    Full COMPAQ Dual-Mode Monitor Support 2-4
  Functional Blocks 2-4
    Graphics Controller 2-5
    Video RAM 2-5
    Attribute Controller 2-5
    Video Output Logic 2-5
    Control and Status Registers 2-6
    Master Clock Selector 2-6
    Sequencer 2-8
    CRT Controller 2-8
  Operating Modes 2-8
    Text Mode 2-9
    Compatible Color Graphics Mode 2-10
    Enhanced Graphics Mode 2-10
2.2 BLOCK DIAGRAMS AND THEIR DESCRIPTIONS 2-12
  Text Mode 2-12
    Setup 2-12
    Operation 2-14
      Character Byte 2-16

TABLE OF CONTENTS (Continued)

Operation (Continued)

Attribute Byte 2-16
Moving the Character and Its Attribute to the CRT Screen 2-18
Timing 2-22
Compatible Color Graphics Mode 2-23
Setup 2-23
Operation (BIOS Modes 4 and 5) 2-24
Data Flow from System Bus to Video RAM Planes 2-25
Data Flow from Video RAM Planes to the CRT Screen 2-25
Operation (BIOS Mode 6) 2-32
Enhanced Graphics Mode 2-34
Write Mode 0 2-36
Data Input to Rotator 2-37
Read Mode 0 2-43
Display Cycle 2-45
Read Mode 1 2-47
Setup 2-50
Operation 2-50
Write Mode 1 2-51
Write Mode 2 2-51
Dual-Mode Support 2-51
Lightpen 2-54
Operation 2-54
2.3 POWER-ON SELF-TEST CODES 2-57

Chapter 2
THEORY OF OPERATION

This chapter describes the theory of operation of the COMPAQ Enhanced Color Graphics Board. The description proceeds from an overview to detailed functional block diagrams and power-on self-test codes for the board.

The overview describes the special features and the logical organization of the board, as well as the board's three major functions (operating modes).

The block diagrams describe in detail each of the three operating modes. Each mode has its own block diagram, which is described according to its functional blocks. Each block diagram is integral to an understanding of an operating mode description. It is recommended that you study each diagram and then read the supporting text.

The power-on self-test codes describe errors that may occur and their probable causes.

2.1 OVERVIEW

Features

The COMPAQ Enhanced Color Graphics Board has a number of special features:

- Color palette
- Video random-access memory (RAM) planes
- Modifiable text fonts
- Logical windowing
- Smooth scrolling and panning
- Split screen
- Full COMPAQ Dual-Mode Monitor support

Color Palette

The COMPAQ Enhanced Color Graphics Board makes available a color palette of 64 colors for display. The palette is composed of a collection of 16 registers, which define the palette colors. These 6-bit registers allow simultaneous display of 16 colors from the palette of 64. The palette provides an extensive selection of colors without the need for increasing video memory size or system processing time.

Although the color palette is available in both 200- and 350-scan line display modes, only the 350-scan line mode allows the COMPAQ Color Monitor or a compatible enhanced color monitor to display the full 64-color palette.

Video RAM Planes

The COMPAQ Enhanced Color Graphics Board contains 256 KB of video RAM organized as four planes of 64 KB each. The planes are designated plane 0, plane 1, plane 2, and plane 3. Depending on the operating mode (text, compatible color graphics, or enhanced color graphics), these planes will be configured accordingly. Figure 2-1 shows the planes.

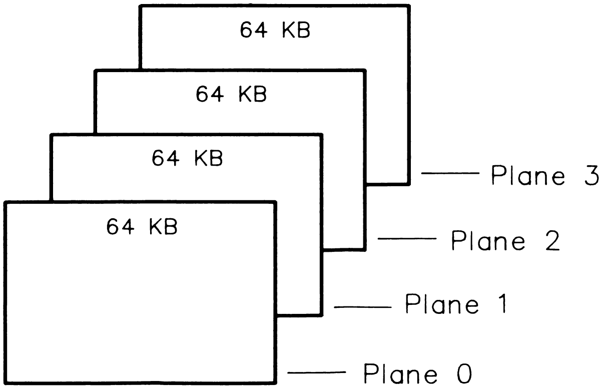


Figure 2-1. Video RAM Planes

Modifiable Text Fonts

Multiple text fonts are stored in the video basic input/output system (BIOS) read-only memory (ROM). When a text mode is entered, the selected font patterns are copied into video RAM plane 2. The patterns then can be modified directly, or an entirely new font pattern can be loaded by means of a video BIOS call.

In the text mode, as many as four font patterns (or, simply, fonts) can reside in video RAM plane 2. Any two of these fonts can be displayed simultaneously, by loading a bit in the attribute byte that specifies the font for any particular character. The remaining two fonts are controlled by the Character Font Select register in the sequencer.

Logical Windowing

Because video memory on the board stores more information than can be displayed at one time on a monitor, a logical windowing feature is provided. Logical windowing allows either text or graphical data displayed on the screen to have an arbitrary starting address in screen memory other than the default offset of 0. Changing the starting address in row increments allows a window on the data to be effectively scrolled without the inefficiency of having to copy blocks of data to and from the screen memory.

Logical line lengths longer than the physical width of the screen are also possible. This feature allows horizontal panning of data in screen memory by simply changing the starting address in character increments. The number of columns that can be panned is a function of how much the logical line length exceeds the physical screen width.

Smooth Scrolling and Panning

The smooth scrolling (vertical movement) feature is an extension to logical windowing. Instead of changing the position of the logical window in whole-row amounts, the next row of information can be brought onto the screen one raster line at a time. This process is often used to eliminate the undesirable visual effects of "jump" scrolling and provides a more pleasing display.

Smooth scrolling occurs when software increments the Preset Row Scan register in conjunction with the Start Address Low and Start Address High registers located in the cathode-ray tube (CRT) controller. These registers are typically updated prior to the vertical retrace period.

The smooth panning (horizontal movement) feature allows the display to be moved horizontally in pixel (dot) increments. These features eliminate the irregular, abrupt display movement common with character scrolling. The panning function is controlled by the Pixel Panning register in the attribute controller.

Smooth scrolling and panning are supported in the text and graphics operating modes.

Split Screen

The split-screen feature allows the simultaneous display of two non-contiguous logical windows on the same screen. The CRT Controller Preset Row/Scan register (3x5h, Index = 08h) specifies the horizontal boundary between the two windows.

Full COMPAQ Dual-Mode Monitor Support

In addition to full support for the COMPAQ Enhanced Color Monitor, the COMPAQ Enhanced Color Graphics Board supports the COMPAQ Dual-Mode Monitor. The 200-scan line mode is directly supported by the COMPAQ Dual-Mode Monitor. The 350-scan line mode is supported by means of COMPAQ-unique circuitry, which allows EGA software (color and monochrome) to function on the COMPAQ Dual-Mode Monitor.

Functional Blocks

The COMPAQ Enhanced Color Graphics Board has eight major functional blocks:

- Graphics controller
- Video RAM
- Attribute controller
- Video output logic
- Control and status registers
- Master clock selector
- Sequencer
- CRT controller

Each functional block is described in the following paragraphs.

Graphics Controller

The graphics controller accepts bit patterns from memory and manipulates the data according to the current graphics mode. It also contains software-accessible color-comparison registers to facilitate color-fill operations. All data passing between the system bus and the video memory is formatted by the graphics controller.

Video RAM

The 256 KB of video memory are divided into four 64-KB planes (planes 0, 1, 2, and 3). Their configuration and addressing can be altered under software control. For example, in the text (color) operating mode, plane 0 is all even-numbered addresses, starting at B8000h, and plane 1 is all odd-numbered addresses, starting at B8001h. Planes 2 and 3 are not software accessible. For more detailed information, refer to Section 2.2, Block Diagrams and Their Descriptions.

Attribute Controller

The attribute controller handles the attribute byte that specifies the colors along with blinking and underlining attributes. It contains the logic used to perform smooth panning and also contains registers for the color palette.

Video Output Logic

Depending on the BIOS mode, the six video output lines from the color palette in the attribute controller can have different interpretations. These video signals pass through the video output logic for adaptation to the COMPAQ Dual-Mode Monitor, when attached. From the video output logic, the signals take two paths: one to the internal monitor circuitry, and the other to the Special Features connector and external monitor connector.

Table 2-1 lists the the various interpretations of the video signals for each BIOS mode. These BIOS modes are described in Chapter 5, BIOS.

Table 2-1. BIOS Modes and Their Corresponding Video Outputs

BIOS Mode	Screen Format, Number of Colors	Video Output
0	40 x 25 text, 16 colors	RGBI
1	40 x 25 text, 16 colors	RGBI
2	80 x 25 text, 16 colors	RGBI
3	80 x 25 text, 16 colors	RGBI
4	320 x 200 graphics, 4 colors	RGBI
5	320 x 200 graphics, 4 colors	RGBI
6	640 x 200 graphics, 2 colors	RGBI
7	80 x 25 text, monochrome	VI
D	320 x 200 graphics, 16 colors	RGBI
E	640 x 200 graphics, 16 colors	RGBI
F	640 x 350 graphics, monochrome	VI
10	640 x 350 graphics, 16/64 colors	RrGgBb
0*	40 x 25 text, 16/64 colors	RrGgBb
1*	40 x 25 text, 16/64 colors	RrGgBb
2*	80 x 25 text, 16/64 colors	RrGgBb
3*	80 x 25 text, 16/64 colors	RrGgBb

Legend:

RGBI = Red, green, blue, and intensity signals for a 16-color monitor.

RrGgBb = Primary and secondary red, green, and blue signals for enhanced color monitors.

VI = Video and intensity signals for a monochrome monitor.

Control and Status Registers

The Control and Status registers of the board provide a means of configuring and sensing the signals that interface the board with the system bus, lightpen, configuration switches, the Special Features connector, and monitors. Additional status registers, such as the Environment register, return information to the software about board capabilities. For detailed information on the Control and Status registers, refer to Chapters 3 and 4.

Master Clock Selector

Selection of the master clock depends on the desired screen resolution and type of monitor attached. During the Set Mode BIOS call, the primary selection of the master clock is made by outputting the selection code to bits <3..2> of the Control and Status Miscellaneous Output register, as shown in Figure 2-2.

Note: Always halt the sequencer before changing the master clock selection. (See Chapter 4, Section 4.3, Sequencer Reset Register.)

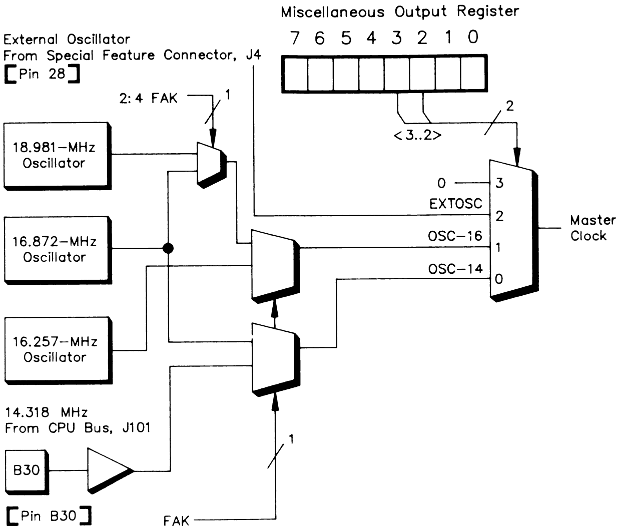

Miscellaneous Output Register
7 6 5 4 3 2 1 0
External Oscillator
From Special Feature Connector, J4
[Pin 28]
2:4 FAK
1
2
<3..2>
18.981-MHz Oscillator
0
3
EXTOSC
2
Master Clock
16.872-MHz Oscillator
OSC-16
1
OSC-14
0
16.257-MHz Oscillator
14.318 MHz
From CPU Bus, J101
B30
1
[Pin B30]
FAK

Figure 2-2. Block Diagram of the Master Clock Selection

When the 350-scan line mode is displayed, the timing characteristics of the COMPAQ Dual-Mode Monitor differ from those of other monitors. These differences are transparently adapted by hardware signals 2:4FAK and FAK that alter the actual master clock selected.

Table 2-2 lists the master clock frequencies.

Table 2-2. Master Clock Frequencies

Display Resolution	Default Software Selection	Alternate Selection if COMPAQ Dual-Mode Monitor Is Active
Enhanced Color:		
640 x 350	16.257 MHz	16.872 MHz
640 x 200	14.318 MHz	14.318 MHz
RGBI Color:		
640 x 200	14.318 MHz	14.318 MHz
Compatible Monochrome (See Note):		
720 x 350	16.257 MHz	18.981 MHz
640 x 350	14.318 MHz	16.872 MHz

Note: The only monochrome monitor supported by the COMPAQ Enhanced Color Graphics Board is the COMPAQ Dual-Mode Monitor; thus, the default software selection frequencies are not output.

Sequencer

The sequencer controls all video memory timing signals including memory refresh. It also arbitrates video memory access between the system bus and the display. For detailed information on the Sequencer registers, refer to Chapters 3 and 4.

CRT Controller

The CRT controller generates the video RAM addresses. It also generates the CRT interface signals, such as horizontal sync (HS) and vertical sync (VS). It contains the lightpen interface and is responsible for smooth scrolling and split-screen functions. For detailed information on the CRT Controller registers, refer to Chapters 3 and 4.

Operating Modes

The three operating modes of the COMPAQ Enhanced Color Graphics Board are the 1) text mode, 2) compatible color graphics mode, and 3) enhanced graphics mode. Each of these operating modes has specific BIOS modes available that offer different screen formats and support different types of monitors. Table 2-3 gives the operating modes and their corresponding BIOS modes.

Table 2-3. Operating Modes and Their Corresponding BIOS Modes and Formats

Operating Mode	BIOS Mode	Description
Text:		
	0	40 x 25 text, 16 colors
	1	40 x 25 text, 16 colors
	2	80 x 25 text, 16 colors
	3	80 x 25 text, 16 colors
	7	80 x 25 text, monochrome (See Note 1)
	0*(See Note 2)	40 x 25 text, 16/64 colors
	1*	40 x 25 text, 16/64 colors
	2*	80 x 25 text, 16/64 colors
	3*	80 x 25 text, 16/64 colors
Compatible Color Graphics:		
	4	320 x 200 graphics, 4 colors
	5	320 x 200 graphics, 4 colors
	6	640 x 200 graphics, 2 colors
Enhanced Graphics:		
	D	320 x 200 graphics, 16 colors
	E	640 x 200 graphics, 16 colors
	F	640 x 350 graphics, monochrome (See Note 1)
	10	640 x 350 graphics, 16/64 colors

Notes: 1. Monochrome BIOS Modes 7 and F provide three intensities.
2. The "*" is a standard notation for four particular BIOS modes in which text is displayed in a high-resolution format.

Text Mode

Text mode formats the screen into an array of character positions, typically 40 or 80 columns of 25 rows. Each displayed character has two associated bytes in video memory: a character byte and an attribute byte. Therefore, 4000 bytes of video memory support one page (or screen) of an 80 x 25 text mode (80 x 25 = 2000 x 2 bytes = 4000 bytes). Character bytes contain the 8-bit ASCII code for a character. Following the character byte in memory is the attribute code for that character. Bit fields within this byte define the displayed character's selected attributes, such as color, blink, background, intensity, and character font.

Table 2-4 lists the text BIOS modes, their respective descriptions, and supported monitors. BIOS Mode 0 and Mode 1 are identical and exist for compatibility with software that enables the CGA color-burst signal. The same is true for Modes 2 and 3, for Modes 0* and 1*, and for Modes 2* and 3*.

Table 2-4. Text BIOS Modes, Their Descriptions, and Monitors Supported

		Monitors Supported		
BIOS Mode	Description	COMPAQ Color	COMPAQ Dual-Mode	RGBI Color
0	40 x 25 text, 16 colors	X	X	X
1	40 x 25 text, 16 colors	X	X	X
2	80 x 25 text, 16 colors	X	X	X
3	80 x 25 text, 16 colors	X	X	X
7	80 x 25 text, monochrome	-	X	-
0*	40 x 25 text, 16/64 colors	X	X	-
1*	40 x 25 text, 16/64 colors	X	X	-
2*	80 x 25 text, 16/64 colors	X	X	-
3*	80 x 25 text, 16/64 colors	X	X	-

Legend:
X = Supported
- = Not supported

Compatible Color Graphics Mode

Most graphics software written for the COMPAQ Video Display Controller Board (VDU) and other controllers compatible with a CGA will run on the COMPAQ Enhanced Color Graphics Board in the compatible graphics mode. This mode allows software addressibility and control of individual pixels in 320 x 200 and 640 x 200 resolution. Although a single page of 32 KB of video memory is allocated, only 16 KB is used beginning at address B8000h. The 320 x 200 resolution has 2 bits per pixel, and the 640 x 200 resolution has 1 bit per pixel, implying a selection of 4 colors and 2 colors, respectively.

Table 2-5 lists the video BIOS modes associated with the compatible color graphics mode. The table also includes the supported monitors. Mode 4 and Mode 5 are identical and exist for compatibility with software that enables the CGA color-burst signal. (Mode 4 signifies that the color-burst signal on the composite video output is enabled. Composite video output is not available on this board.)

Table 2-5. Compatible Color Graphics BIOS Modes, Their Descriptions, and Monitors Supported

		Monitors Supported		
BIOS Mode	Description	COMPAQ Color	COMPAQ Dual-Mode	RGBI Color
4	320 x 200 graphics, 4 colors	X	X	X
5	320 x 200 graphics, 4 colors	X	X	X
6	640 x 200 graphics, 2 colors	X	X	X

Legend:
X = Supported

Enhanced Graphics Mode

In addition to supporting the existing BIOS modes of the COMPAQ Video Display Controller Board, the CGA, and the MDA, the board adds four enhanced graphics BIOS modes: Modes D, E, F, and 10. Table 2-6 lists these modes.

Enhanced graphics modes allow fast manipulation of graphical screen images by an architecture that places the four color planes in parallel with a hardware assist that rotates, masks, and performs Boolean operations on as many as 32 bits in a single central processing unit (CPU) Write cycle. For the four enhanced graphics modes, 64 KB of video memory (for each of the four planes) is software accessible from address A0000h to AFFFFh.

Modes D and E provide 320 x 200 and 640 x 200 screen resolutions, respectively, in 16 colors. Modes D and E can be displayed on all three types of supported monitors.

Table 2-6. Enhanced Graphics BIOS Modes, Their Descriptions, and Monitors Supported

		Monitors Supported		
BIOS Mode	Description	COMPAQ Color	COMPAQ Dual-Mode	RGBI Color
D	320 x 200 graphics, 16 colors	X	X	X
E	640 x 200 graphics, 16 colors	X	X	X
F	640 x 350 monochrome, 3 intensities	-	X	-
10	640 x 350 graphics, 16/64 colors*	X	X	-

Legend:
X = Supported
- = Not Supported

Note: On a COMPAQ Dual-Mode Monitor, Modes D, E, and 10 display 15 intensities.

Mode F provides 640 x 350 resolution on a COMPAQ Dual-Mode Monitor in three monochrome intensities, plus a pixel-blinking attribute. This mode allows software written for an EGA configured with a compatible monochrome monitor to run on the COMPAQ Enhanced Color Graphics Board configured with a COMPAQ Dual-Mode Monitor.

Mode 10 provides a 640 x 350 resolution in either 16 out of 64 colors on a COMPAQ Color Monitor or 15 intensities on a COMPAQ Dual-Mode Monitor.

2.2 BLOCK DIAGRAMS AND THEIR DESCRIPTIONS

Each of the three operating modes of the COMPAQ Enhanced Color Graphics Board has its own block diagram. The layout of each diagram reflects how the software configures the board for a particular mode. The way data flows between the blocks and how the blocks are logically connected on a block diagram depends on the mode in which the board is operating.

The block diagrams and descriptions in this section are intended to provide a general understanding of data flow through the COMPAQ Enhanced Color Graphics Board.

The following descriptions address the text mode, the compatible color graphics mode, and the enhanced graphics mode. The descriptions of each operating mode are generally organized in two major parts: setup and operation.

For the purpose of reducing repetitious information, descriptions of the concepts common to the three operating modes are not repeated. You are referred to the operating mode in which the concept was initially described.

Text Mode

Prior to operating in any mode, the board must be initialized. To initialize the board, proper data for the selected BIOS mode from video ROM tables must be loaded into the various programmable registers.

Setup

Setup starts with power-on, and the process is as follows:

Process

1. Power is applied to the system.

2. The CPU executes a power-on (diagnostic) self-test, which is resident in the system ROM.

3. The CPU (that is, the program) scans for option ROMs.

4. The program recognizes the COMPAQ Enhanced Color Graphics Board ROM.

5. The program transfers control from the system to the video BIOS ROM.

6. The video BIOS ROM sets the board hardware to a known state.

7. The video BIOS ROM performs a self-test by:
   a) Checking COMPAQ-specific registers
   b) Checking for vertical and horizontal sync (checks sync timing using system clock)
   c) Checking for video presence at output of the attribute controller
   d) Checking video RAM
   e) Checking for illegal switch settings.

   If the self-test detects a problem, it reports the error with one long and two short beeps at 1200 Hz, and writes the error code to I/O diagnostic ports 084h and 085h. Port 084h receives the two-digit error code. (See Section 2.3, Power-On Self-Test Codes). Port 085h receives the device code.

8. The Set Mode BIOS call is issued, and the board is initialized in the mode designated by the setting of configuration switch bank SW1. (For more detailed information on SW1 settings, refer to Chapter 6, Section 6.2, Switches and Jumpers.)

   Example:

   To initialize the board for BIOS Mode 3* (80 x 25 color, 350-scan line format), set SW1 positions 1, 2, 3, and 4 to OFF, ON, ON, OFF, respectively.

The board always starts up in text mode. Any time it is necessary to change from text mode to graphics mode (for example, while running an applications program using graphical illustrations intermixed with text), the Set Mode BIOS call is issued. The call loads all programmable registers with data from tables located in BIOS ROM (which table is loaded depends on which mode you have selected). The Set Mode call loads (initializes) the:

- Font (RAM plane 2)
- Odd/even RAM plane setup
- Palette (located in the attribute controller)
- Sequencer (which performs read/write timing)
- CRT controller (which generates RAM plane addresses)
- Master clock

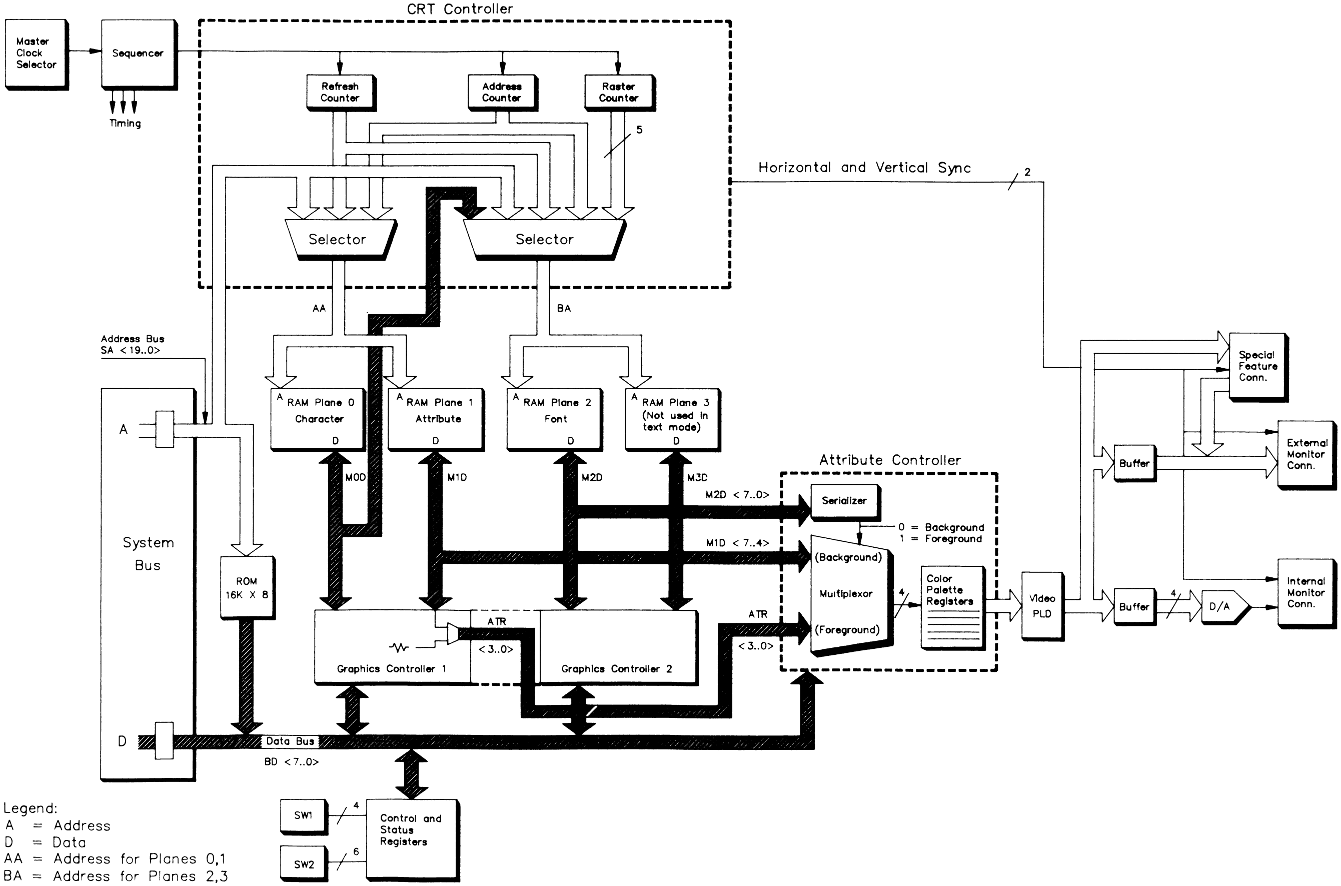
The BIOS ROM tables, which give values for all programmable registers for a given BIOS mode, are provided in Chapter 4, Section 4.6, Mode-Specific Register Values.

Operation

Once the board is initialized for text mode operation, text can move from the system bus through the board to the CRT screen.

In order to trace the flow of data, study the diagram shown in Figure 2-3. To set up a hypothetical exercise, assume you have selected the text operating mode and BIOS Mode 3*. Assume also that you want to write a capital letter "A" (character byte = 41h) at address B8000h, which corresponds to the upper-left corner of the CRT screen. The "A" is to be white with a black background (attribute byte = 07h).

The following text describes how that character is written to the RAM planes (each plane is a portion of video memory), then moved to the screen. It also describes how attribute data determine the background and foreground characteristics of the character. Notice that the character byte and the attribute byte of data appear on the system bus, which is on the left side of Figure 2-3.


CRT Controller
Master Clock Selector
Sequencer
Timing
Refresh Counter
Address Counter
Raster Counter
5
Horizontal and Vertical Sync
2
Selector
Selector
AA
BA
Address Bus
SA <19..0>
A
System Bus
D
RAM Plane 0
Character
RAM Plane 1
Attribute
RAM Plane 2
Font
RAM Plane 3
(Not used in text mode)
MOD
M1D
M2D
M3D
M2D <7..0>
M1D <7..4>
ATR
<3..0>
ATR
<3..0>
ROM
16K X 8
Graphics Controller 1
Graphics Controller 2
Data Bus
BD <7..0>
Attribute Controller
Serializer
0 = Background
1 = Foreground
(Background)
Multiplexor
(Foreground)
4
Color Palette Registers
Video PLD
Buffer
Buffer
4
D/A
Special Feature Conn.
External Monitor Conn.
Internal Monitor Conn.
SW1
4
SW2
6
Control and Status Registers

Legend:
A = Address
D = Data
AA = Address for Planes 0,1
BA = Address for Planes 2,3

Figure 2-3. Block Diagram of the Text Mode (BIOS Modes 0, 1, 2, 3, 7, 0*, 1*, 2*, and 3*)

Character Byte

The system bus interface handles 8-bit data (the character byte and the attribute byte) and 20-bit addresses. To write a data byte to video memory (planes 0..3), the system issues a Memory Write (MEMW-) signal. This signal moves the character byte, which had been placed on the data bus, to video RAM plane 0. (The attribute byte is handled in the next Memory Write cycle). The process is outlined below:

Process

1. The system places the character byte (41h -- capital letter "A") on the data bus.
2. At the same time, the 20-bit memory address that stores the "A" in plane 0 (B8000h) appears on the address bus. The CRT controller receives the address and routes it to plane 0. This address specifies where in plane 0 the character is to be stored.
3. The character byte on the data bus is received by graphics controller 1 and is sent to plane 0. The character "A" is stored in memory at the address that appears on the plane 0 address bus. The memory address, B8000h in this case, corresponds to the upper-left corner of the screen. Note the direct correlation between the storage location of "A" in plane 0 and the location of "A" on the display.

Attribute Byte

The attribute byte is placed on the system bus immediately after the character byte. The process is outlined below:

Process

1. A MEMW- signal is issued.
2. The attribute byte is received by graphics controller 1.
3. The attribute byte is then sent to plane 1. The system address byte specified by the bus (which is B8001h, one number higher than the character address).

The attribute byte in text mode is defined in Figure 2-4.

Blink or ATR3	ATR2	ATR1	ATR0	Font or ATR3	ATR2	ATR1	ATR0
Background				Foreground			

Figure 2-4. Definition of the Attribute Byte

In this hypothetical exercise, to select the background color black and the foreground (character) color white, a 07h is entered in the attribute byte. (The 07h is used to select colors from the palette later in the process.)

Note that at the end of two Memory Write cycles, the character is stored in plane 0 and the attribute is stored in plane 1.

Moving the Character and Its Attribute to the CRT Screen

The character that appears on the screen is taken from the character fonts stored in plane 2. As many as four character sets (fonts) can be stored in plane 2. To select the character "A" from one of the stored fonts, the appropriate address must be generated. The format of the font address is shown in Figure 2-5.

<table>
<tr><th colspan="16">Bits</th></tr>
<tr><th>15</th><th>14</th><th>13</th><th>12</th><th>11</th><th>10</th><th>9</th><th>8</th><th>7</th><th>6</th><th>5</th><th>4</th><th>3</th><th>2</th><th>1</th><th>0</th></tr>
<tr><td>0</td><td>0</td><td>0</td><td>0</td><td>1</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>1</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr>
<tr><td colspan="2">Character Font Select</td><td>Always 0</td><td colspan="8">Character (ASCII 41h from plane 0)</td><td colspan="5">Scan Line (from CRT controller)</td></tr>
</table>

Figure 2-5. Format of the Font Address in Plane 2

The four components of this address, as shown in Figure 2-5, appear at plane 2 simultaneously. The sources for the address components are derived as follows:

Bits <15..14>:	These bit values are set in the Character Font Select register in the sequencer block. (For possible values, see Chapter 4, Section 4.3, Sequencer Registers). These bits determine one of four possible character fonts in which the "A" is to appear.
Bit <13>:	This bit value is always forced to 0.
Bits <12..5>:	These bit values specify which character comes out of the font. The bits are sent as data from plane 0 through the CRT controller to plane 2.
Bits <4..0>:	These bit values are from the raster counter in the CRT controller. These bits determine the raster line to be displayed (scan lines 0 through 13 in the 8 x 14 font). Note that each character in the font is stored, in this case, in a matrix 8 bits wide and 14 rows (scan lines or rasters) deep, which is shown in Figure 2-6. Each scan line is read out 8 bits (one row) at a time. The matrix is actually 32 rows deep with only the first 14 rows containing a font character.

8 x 14 Font

Raster Line									Hexadecimal Representation of the Character "A"	Address in Plane 2
0									00h	0820h
1									00h	0821h
2				■					10h	0822h
3			■	■	■				38h	0823h
4		■	■		■	■			6Ch	0824h
5	■	■				■	■		C6h	0825h
6	■	■				■	■		C6h	0826h
7	■	■	■	■	■	■	■		FEh	0827h
8	■	■				■	■		C6h	0828h
9	■	■				■	■		C6h	0829h
10	■	■				■	■		C6h	082Ah
11									00h	082Bh
12									00h	082Ch
13									00h	082Dh

Figure 2-6. Matrix for Character "A" in the 8 x 14 Font

For our hypothetical exercise, the font address, as was shown in Figure 2-5, yields 0820h--the address of scan line 0 of the pattern for "A", as stored in plane 2.

To summarize, the character "A" is sent from the system bus through graphics controller 1 to plane 0, and then from plane 0 as an address to font plane 2. The attribute byte has followed a similar path from the system bus to graphics controller 1 to plane 1.

The timing is such that when a raster line is read out of the font plane (on the M2D lines) and appears at the attribute controller, the attribute byte from plane 1 for the character "A" also appears at the attribute controller. The upper 4 bits (background attribute) appear on the M1D lines, bits <7..4>. The lower 4 bits (foreground attribute) appear on the ATR lines, bits <3..0>. Handling of the attribute byte by the attribute controller is described in the following process:

Process

1. The attribute is read from plane 1 at the same time the character is read from plane 0.

2. The attribute appears at the input of the background/foreground attribute multiplexor in the attribute controller.

3. The character scan line (8 bits) appears at the input to the serializer.

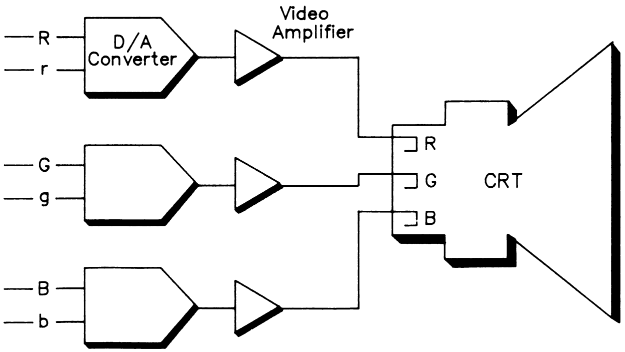
4. The foreground is selected when the output of the serializer is a 1. Background is selected when the output of the serializer is a 0. The ones and zeros from the serializer turn the dots on and off, respectively.

5. The output of the background/foreground multiplexor is 4 bits, which become an address for the color palette. The color palette has 16 registers of 6 bits each. For our hypothetical exercise, the attribute byte has specified a background color of black (color code 0) and a foreground color of white (color code 7). Table 2-7 lists a palette with registers programmed for BIOS Mode 3*.

Table 2-7. Color Palette Values with Registers Programmed for Mode 3* (80 x 25 Text)

Color Code	Attribute Bits 3210	Palette Values rgbRGB	Display Color
0	0000	000000	Black
1	0001	000001	Blue
2	0010	000010	Green
3	0011	000011	Cyan
4	0100	000100	Red
5	0101	000101	Magenta
6	0110	010100	Brown
7	0111	000111	White
8	1000	111000	Gray
9	1001	111001	Light Blue
10	1010	111010	Light Green
11	1011	111011	Light Cyan
12	1100	111100	Light Red
13	1101	111101	Light Magenta
14	1110	111110	Yellow
15	1111	111111	Intense White

6. The color selected and its intensity depend on which bits are set in the color palette. The r, g, b, and R, G, B in the color palette correspond to the r, g, b, and R, G, B lines in the monitor. Color intensity can be selected, as shown in Figure 2-7.



R	r	Red Intensity
0	0	0%
0	1	33% (1/3)
1	0	67% (2/3)
1	1	100%

Figure 2-7. Color Selection in the Monitor

7. Each bit in a color palette register is routed to one of six inputs in the monitor.

8. As the bits are read out of the color palette, they are input to the corresponding lines in the monitor. A particular color is selected by setting the appropriate bit pattern in the color palette. For our hypothetical exercise, the attribute byte (07h) selects black (000000) for background and white (000111) for foreground.

Timing

The master clock provides the time base for the board. For our hypothetical exercise--BIOS Mode 3* with a COMPAQ Color Monitor--the master clock frequency is 16.257 MHz. This clock frequency drives the sequencer.

The sequencer provides all the timing and coordination for any reading and writing to the video RAM planes. It also provides a character clock output to the CRT controller as well as a dot clock output to the attribute controller and graphics controller. The dot clock runs the serializer.

The CRT controller generates all RAM plane addresses and provides horizontal and vertical sync and blanking for the monitor. It also maintains cursor location and type on the screen.

Compatible Color Graphics Mode

The process for initializing the COMPAQ Enhanced Color Graphics Board for the compatible color graphics operating mode is similar to that for the text operating mode. The similarity is in the sequence of register programming.

Setup

Once the Set Mode BIOS call is issued for the graphics mode, the programmable registers are initialized for CGA BIOS Mode 4, 5, or 6.

The compatible color graphics mode of the board provides a screen memory and BIOS-level compatibility with CGA BIOS Modes 4, 5, and 6. Modes 4 and 5 define 320 x 200 resolution, four-color graphics. Mode 6 defines 640 x 200 resolution, two-color graphics.

Configuration of the compatible color graphics mode includes mapping the screen memory at B8000h and the base I/O address at 3D4h. Odd/even memory access of RAM planes 0 and 1 is programmed for 320 x 200 resolution (Modes 4 and 5), whereas only plane 0 is used for the 640 x 200 resolution (Mode 6). Planes 2 and 3 are not used for the compatible color graphics mode. The graphics controller is otherwise programmed to pass unaltered all data between the CPU and the screen memory.

Memory addressing for the compatible color graphics mode is interleaved, unlike the text and enhanced graphics modes. In text and enhanced graphics modes, the address of the last displayed item on one row is immediately followed by the address of the first item on the next sequential row. However, in compatible color graphics mode, the addresses of the scan lines are interleaved, as shown in Figure 2-8. All the even-numbered scan lines are contiguous and begin at B8000h. All the odd-numbered scan lines are contiguous, but begin at BA000h (B8000h plus an offset of 2000h).

Linear addressing (BIOS Modes E, F, 10)

	Column 0		Column 79
Row 0	A0000h	....	A004Fh
Row 1	A0050h	....	A009Fh
Row 2	A00A0h	....	A00EFh
Row 3	A00F0h	....	A013Fh
	⋮		⋮

Interleaved Addressing (BIOS Modes 4, 5, 6)

	Column 0		Column 79
Row 0	B8000h	....	B804Fh
Row 1	BA000h	....	BA04Fh
Row 2	B8050h	....	B809Fh
Row 3	BA050h	....	BA09Fh

Figure 2-8. Comparison of Linear Addressing with Interleaved Addressing

Operation (BIOS Modes 4 and 5)

In the text mode, the data that appeared on the system bus were composed of 1 byte for a text character and 1 byte for the character attribute. In the compatible color graphics mode, each byte controls 4 pixels. With reference to the byte that represents the upper-left corner of the CRT screen, the relationship between bit pairs and pixels is shown in Figure 2-9.

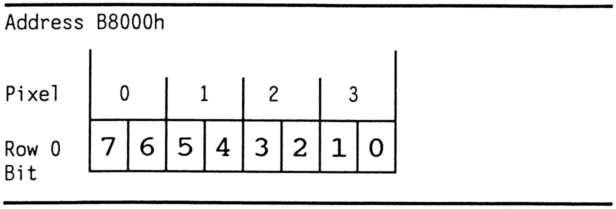


Figure 2-9. Relationship Between Bit Pairs and Pixels in the Compatible Color Graphics Mode (BIOS Modes 4 and 5)

The address of this byte of data (representing the first 4 pixels on the screen) is B8000h. The address of the second byte of data representing the next 4 pixels (fifth through eighth) is B8001h, and so on.

Video memory for BIOS Modes 4 and 5 is organized so that even-numbered bytes (for example, the byte at B8000h) are stored in plane 0 and the odd-numbered bytes (for example, the byte at B8001h) are stored in plane 1. Planes 2 and 3 are not used.

As previously noted, each pixel is controlled by 2 bits. Each bit is an attribute of a particular pixel, and the bits control the pixel's color, as indicated in Table 2-8.

Table 2-8. Colors Selected by Attribute Bits

Attribute Bits		Normal		Intensified	
ATR1	ATR0	Set 1	Set 0	Set 1	Set 0
0	0	Backgnd	Backgnd	Backgnd	Backgnd
0	1	Cyan	Green	Lt. Cyan	Lt. Green
1	0	Magenta	Red	Lt. Magenta	Lt. Red
1	1	White	Brown	Int. White	Yellow

Note: Background = Whatever has been programmed into the Color Palette 0 register (default is black).

The background (black in our exercise) is whatever has been programmed into the Color Palette 0 register. Note that because 2 bits control 1 pixel, only four combinations of addresses are possible. Therefore, only the first 4 (out of 16 possible) palette registers can be addressed. The color set in the palette registers depends on what data were loaded during initialization.

Data Flow from System Bus to Video RAM Planes

In order to trace the flow of data, study the diagram shown in Figure 2-10.

As in the text mode, a byte is written to screen memory from the system bus when:

1. The CPU places the address of the byte on the address bus (A<19..0>).
2. The byte of data is placed on the 8-bit data bus (D<7..0>).
3. The MEMW- signal is asserted.

For the 320 x 200 resolution, four-color BIOS Modes 4 and 5, the byte written by the CPU is directed to plane 0 when its address is even-numbered and to plane 1 when odd-numbered.

Data Flow from Video RAM Planes to the CRT Screen

During the Memory Write cycle, data (1 byte per plane) is read out of plane 0 and plane 1 simultaneously. These 2 bytes appear at the serializers in graphics controller 1.

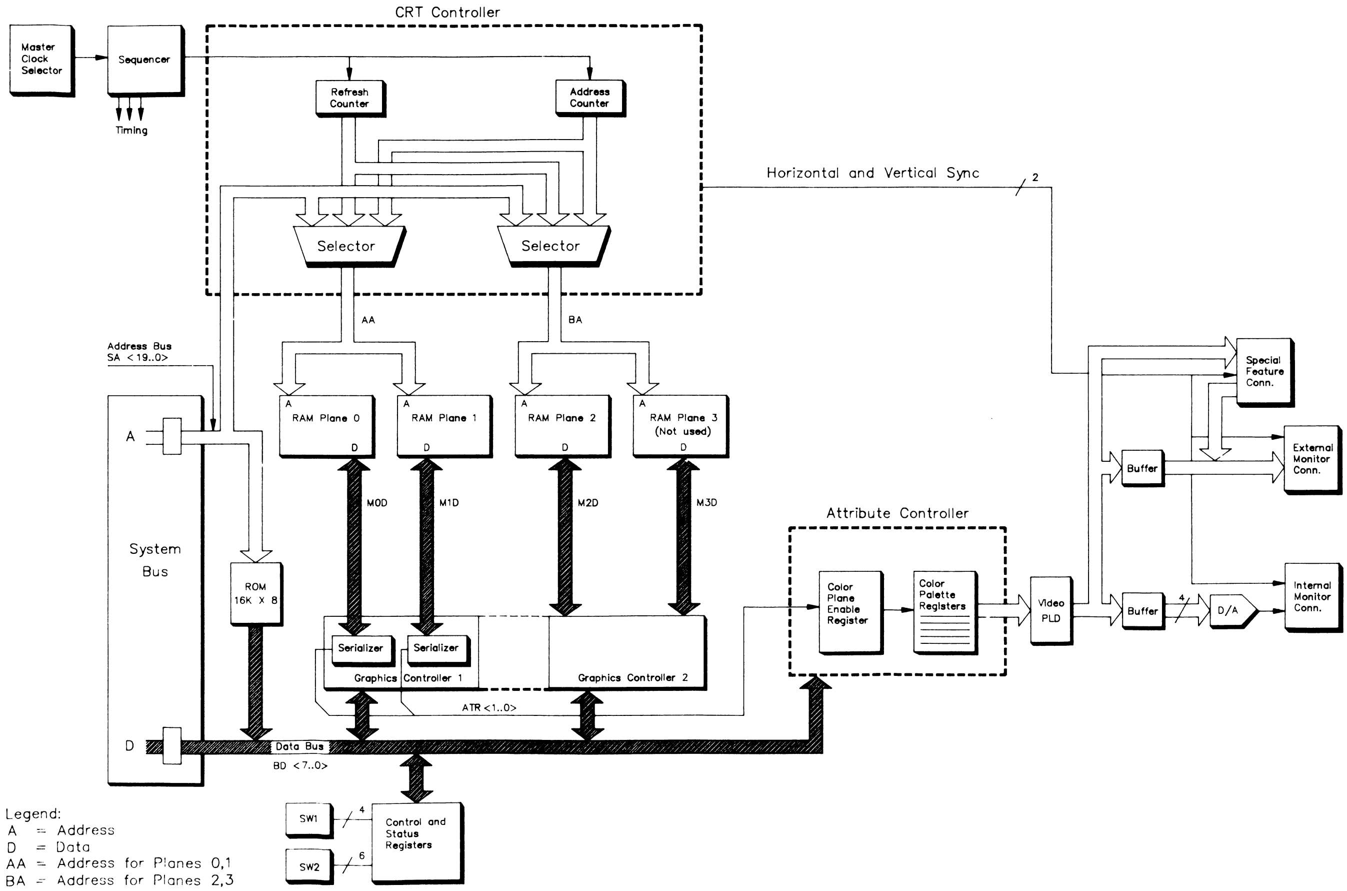

CRT Controller
Master Clock Selector
Sequencer
Timing
Refresh Counter
Address Counter
Horizontal and Vertical Sync
2
Selector
Selector
AA
BA
Address Bus
SA <19..0>
A
RAM Plane 0
D
A
RAM Plane 1
D
A
RAM Plane 2
D
A
RAM Plane 3
(Not used)
D
M0D
M1D
M2D
M3D
System Bus
A
D
ROM
16K X 8
Serializer
Serializer
Graphics Controller 1
Graphics Controller 2
ATR <1..0>
Data Bus
BD <7..0>
Attribute Controller
Color Plane Enable Register
Color Palette Registers
Video PLD
Special Feature Conn.
Buffer
External Monitor Conn.
Buffer
4
D/A
Internal Monitor Conn.
SW1
4
SW2
6
Control and Status Registers
Legend:
A = Address
D = Data
AA = Address for Planes 0,1
BA = Address for Planes 2,3

Figure 2-10. Block Diagram of the Compatible Color Graphics Mode (BIOS Modes 4, 5, and 6)

In the graphics controller, data are serialized 16 bits at a time by a pair of serializers (shift registers) that send their outputs to the attribute controller over the ATR<1..0> lines. These serializers format the data bits, as illustrated in Figure 2-11.

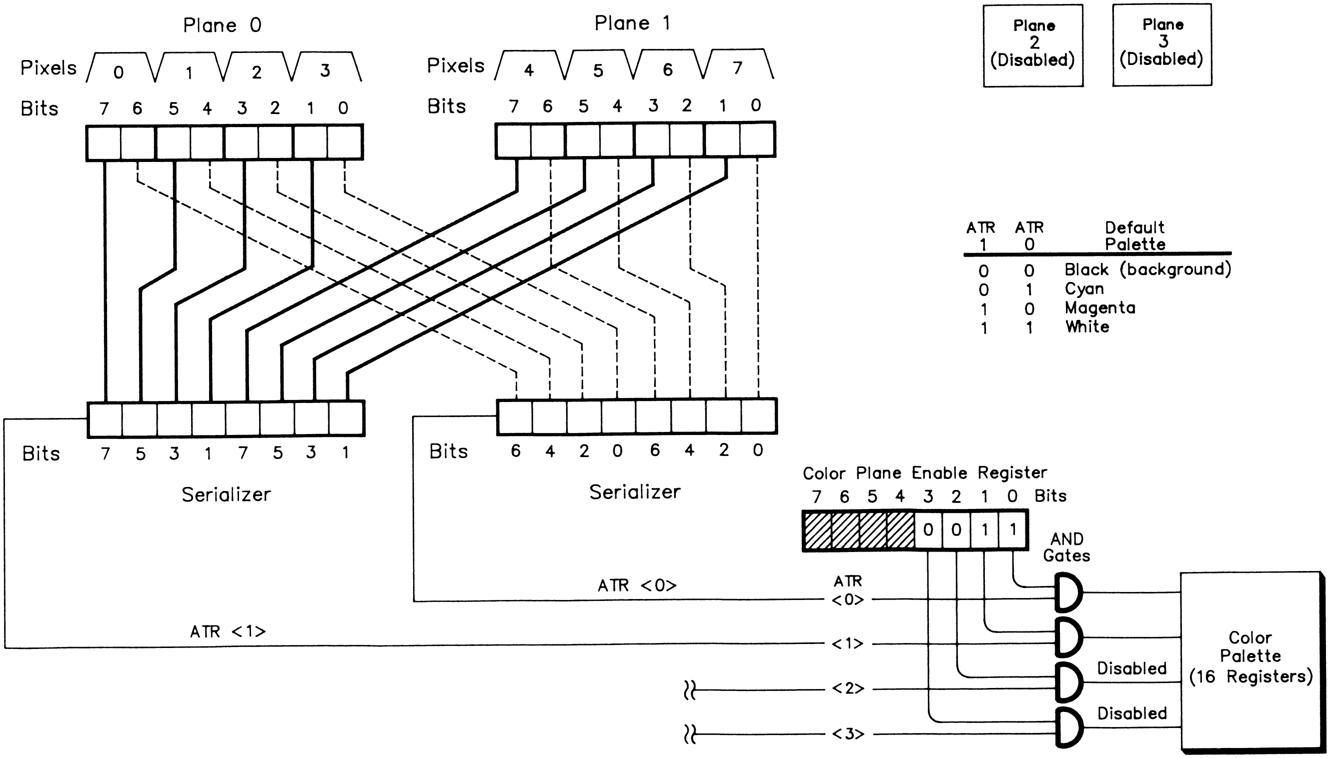

Plane 0
Pixels 0 1 2 3
Bits 7 6 5 4 3 2 1 0
Plane 1
Pixels 4 5 6 7
Bits 7 6 5 4 3 2 1 0
Plane 2 (Disabled)
Plane 3 (Disabled)
ATR 1 ATR 0 Default Palette
0 0 Black (background)
0 1 Cyan
1 0 Magenta
1 1 White
Bits 7 5 3 1 7 5 3 1
Serializer
Bits 6 4 2 0 6 4 2 0
Serializer
Color Plane Enable Register
7 6 5 4 3 2 1 0 Bits
0 0 1 1
AND Gates
ATR <0>
ATR <1>
ATR <0>
<1>
<2>
<3>
Disabled
Disabled
Color Palette (16 Registers)

Figure 2-11. Data Flow of BIOS Modes 4 and 5

The data from planes 0 and 1 are input to the serializers so that all the odd-numbered bits are loaded into one serializer and the even-numbered bits are loaded into the other serializer.

Data are shifted out to the left so that bits <7> (ATR line 1) and <6> (ATR line 0) leave first, followed by bits <5> (ATR line 1) and <4> (ATR line 0), and so on. Serializers are configured in this manner when the Graphics Controller Mode register, bit <5>, is set. Each pair of bits that leaves the serializers is input to the Color Plane Enable register, which is also illustrated in Figure 2-11.

The purpose of the Color Plane Enable register is to form the address that selects a particular color register in the palette. A 0 in any bit position in the Color Plane Enable register disables the respective ATR line. This action, in turn, masks out the corresponding RAM plane and the number of colors is reduced accordingly. ATR lines 2 and 3 are disabled, thereby masking planes 2 and 3.

The output of the Color Plane Enable register becomes the address for the color palette. As previously indicated in Table 2-8, each pair of bits selects a palette register (0..3), which contains a previously loaded bit pattern that represents a certain color.

For example, if the input to the color palette is 0011 and the intensified color-set 1 has been loaded into the palette, the output of the palette is the bit pattern representing intense white.

The number of active bits in the color pallette register vary, depending on the monitor configured with the system.

For RGBI color (200-scan line) operation, the bits in the color palette register have the following meaning:

Bit

7	6	5	4	3	2	1	0
▨	▨	0	I	0	R	G	B

(See Note)

For RrGgBb enhanced color (350-scan line) operation, the palette is programmed as follows:

Bit

7	6	5	4	3	2	1	0
▨	▨	r	g	b	R	G	B

(See Note)

For monochrome (350-scan line) operation, the palette is programmed as follows:

Bit

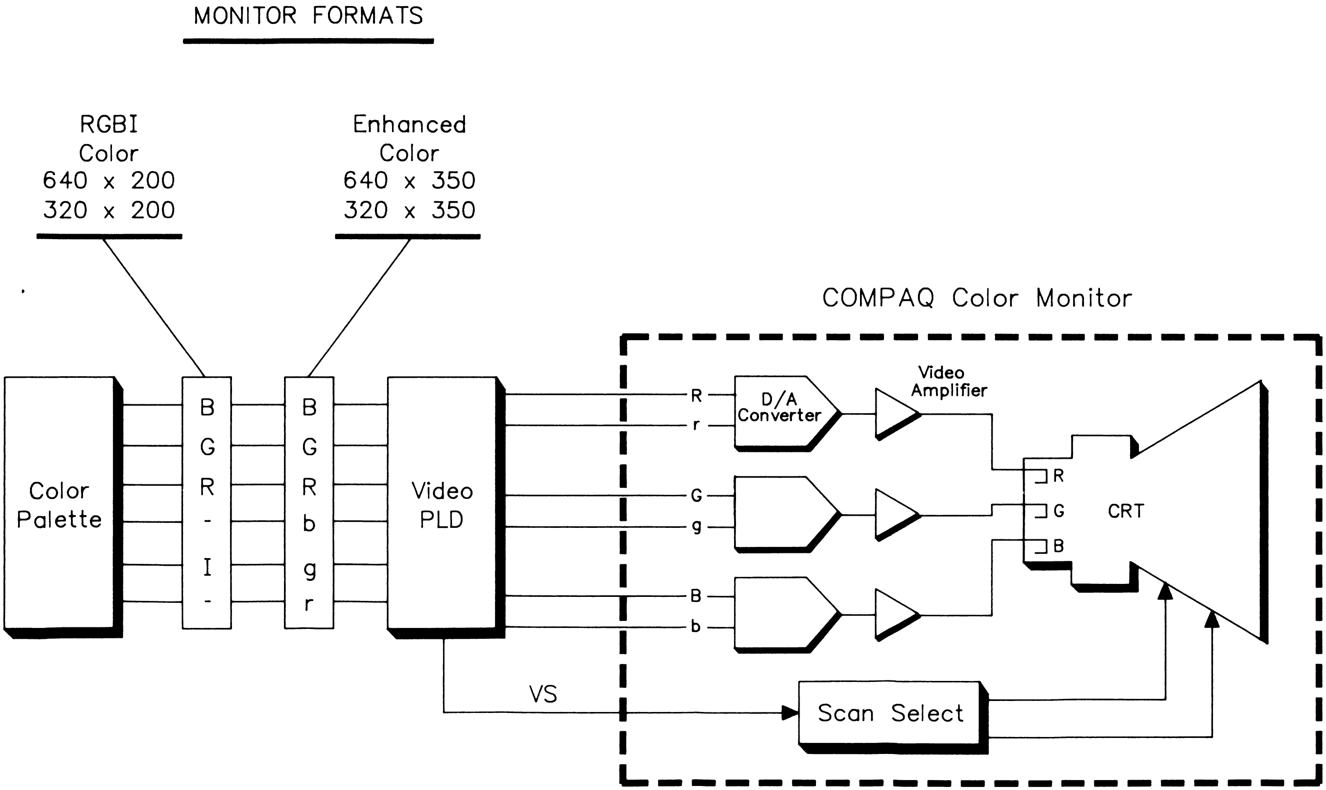
7	6	5	4	3	2	1	0
▨	▨	0	I	V	0	0	0

(See Note)

Note: Shading indicates that bit is not used, and its value should be 0.

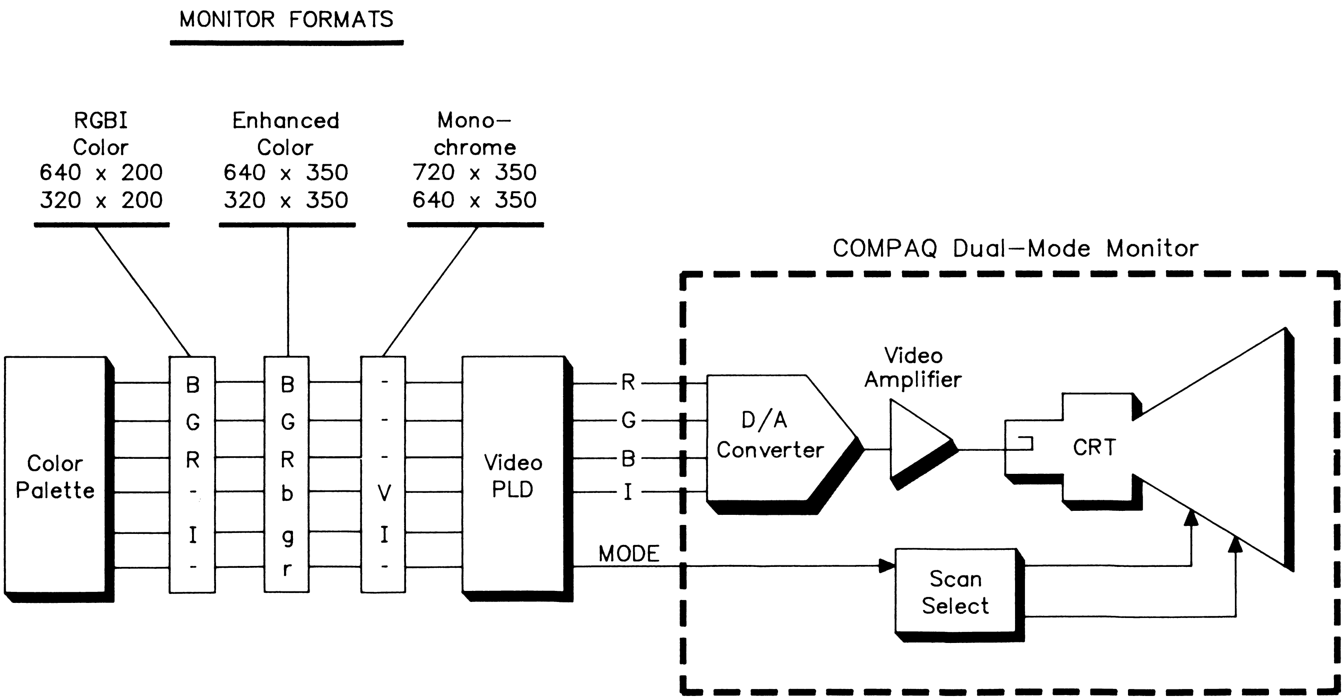
The output of each bit position is routed to the corresponding line in the monitor via the video programmable logic device (PLD). The video PLD is part of the video output logic block. Figure 2-12 illustrates this output for the COMPAQ Color Monitor. Figure 2-13 illustrates this output for the COMPAQ Dual-Mode Monitor.

The six lines coming out of the color palette can be programmed in three different ways, depending on the BIOS mode in operation. This is shown in the illustrations.


MONITOR FORMATS
RGBI Color
640 x 200
320 x 200
Enhanced Color
640 x 350
320 x 350
Color Palette
B
G
R
-
I
-
B
G
R
b
g
r
Video PLD
COMPAQ Color Monitor
R
r
G
g
B
b
D/A Converter
Video Amplifier
R
G
B
CRT
VS
Scan Select

Note: Monitor format determines color palette output to the monitor.

Figure 2-12. Programmable Logic Device for the COMPAQ Color Monitor


MONITOR FORMATS
RGBI Color
640 x 200
320 x 200
Enhanced Color
640 x 350
320 x 350
Mono-chrome
720 x 350
640 x 350
COMPAQ Dual-Mode Monitor
Color Palette
B
G
R
-
I
-
B
G
R
b
g
r
-
-
-
V
I
-
Video PLD
R
G
B
I
MODE
D/A Converter
Video Amplifier
CRT
Scan Select

Note: Monitor format determines color palette output to the monitor.

Figure 2-13. Programmable Logic Device for the COMPAQ Dual-Mode Monitor

The COMPAQ Dual-Mode Monitor recognizes all three types of palette output (monochrome, compatible color, and enhanced color). However, because it has an RGBI interface, an intervening process is necessary to convert the data on six lines (R,r,G,g,B,b) to four lines, or to convert the data on two lines (V, I) to four lines. This conversion process is the function of the video PLD.

When an enhanced color mode (BIOS Modes 10, 0*, 1*, 2*, or 3*) is selected and a COMPAQ Color Monitor or compatible enhanced color monitor is used, the video PLD passes the six signals (R,r,G,g,B,b) from the color palette unaltered. However, if a COMPAQ Dual-Mode Monitor is used, the video PLD receives the 64 color codes from the color palette and merges colors that are close to each other in the color spectrum to produce an output of 16 color codes on four lines.

When a color mode (BIOS Modes 0..6, D, or E) is selected, the video PLD passes without alteration the four-line 16-color input (RGBI) to the monitor.

When a monochrome mode (BIOS Modes 7 or F) is selected and a COMPAQ Dual-Mode Monitor is used, the monochrome video (V) bit is mapped to the R, G, and B lines of the monitor. The intensity (I) bit is mapped to the I line of the monitor. (The COMPAQ Color Monitor does not support this mode.)

Operation (BIOS Mode 6)

Unlike BIOS Modes 4 and 5, in the 640 x 200 2-color mode (Mode 6), all data are written into plane 0, with the outputs of planes 1, 2, and 3 disabled (ignored) by the Color Plane Enable register. The byte of data output from plane 0 goes to a single serializer, as shown in Figure 2-14.

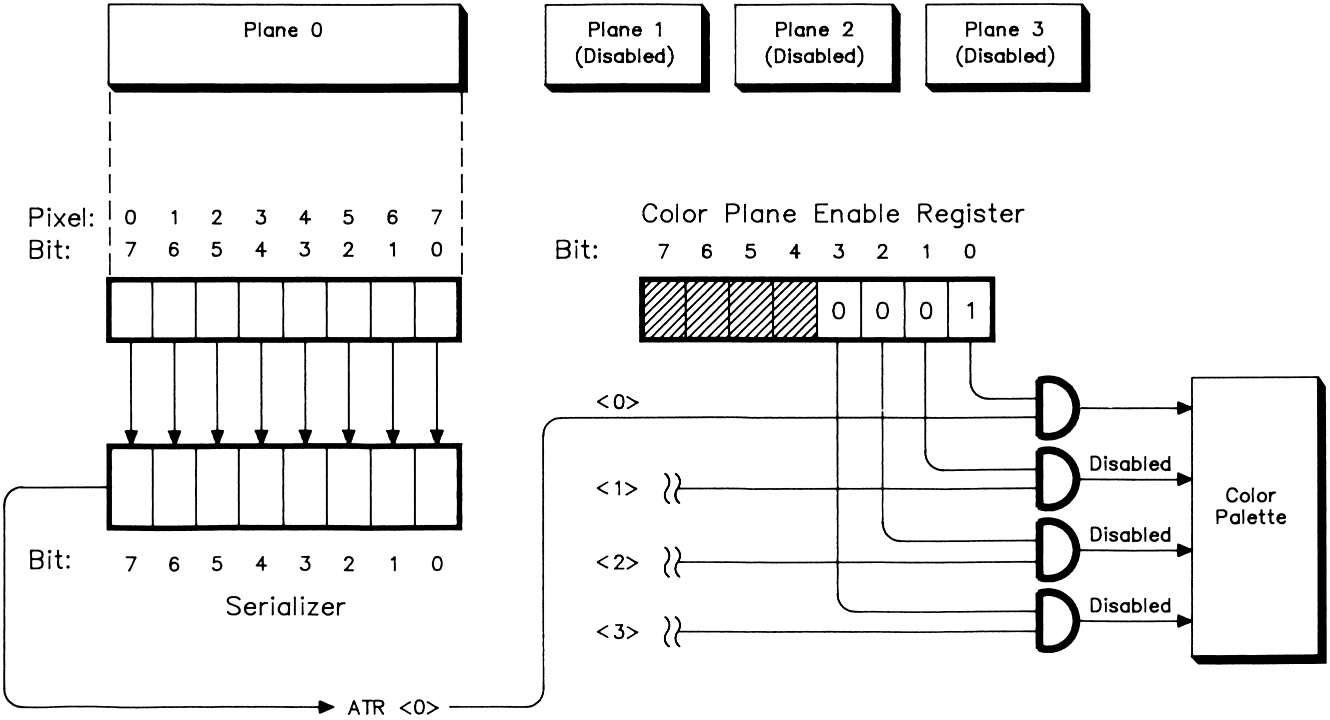

Plane 0
Plane 1 (Disabled)
Plane 2 (Disabled)
Plane 3 (Disabled)
Pixel: 0 1 2 3 4 5 6 7
Bit: 7 6 5 4 3 2 1 0
Color Plane Enable Register
Bit: 7 6 5 4 3 2 1 0
0 0 0 1
<0>
<1>
<2>
<3>
Disabled
Disabled
Disabled
Color Palette
Bit: 7 6 5 4 3 2 1 0
Serializer
ATR <0>

Figure 2-14. Data Flow from BIOS Mode 6

In BIOS Mode 6, 1 bit controls 1 pixel. The byte from plane 0 is shifted to the left out of the serializer starting with bit <7> (pixel 0). The bits from the serializer are output to the color palette from the Color Plane Enable register. The output from the Color Plane Enable register (either a 1 or 0) can select only two colors from the Color Palette (Color 0 register or Color 1 register).

The output from the color palette is routed to the video PLD and is handled as described previously for Modes 4 and 5.

Enhanced Graphics Mode

The enhanced graphics mode consists of three Write modes and two Read modes.

Write Mode 0 allows bit manipulation (for example, ANDing, ORing) of data going from the system to memory. This mode writes individual pixels to the screen.

Write Mode 1 allows the copying of a byte from one screen location to several (or all) other screen locations. Data come from the Read Data latch in the Read logic (in the graphics controller) rather than from the system.

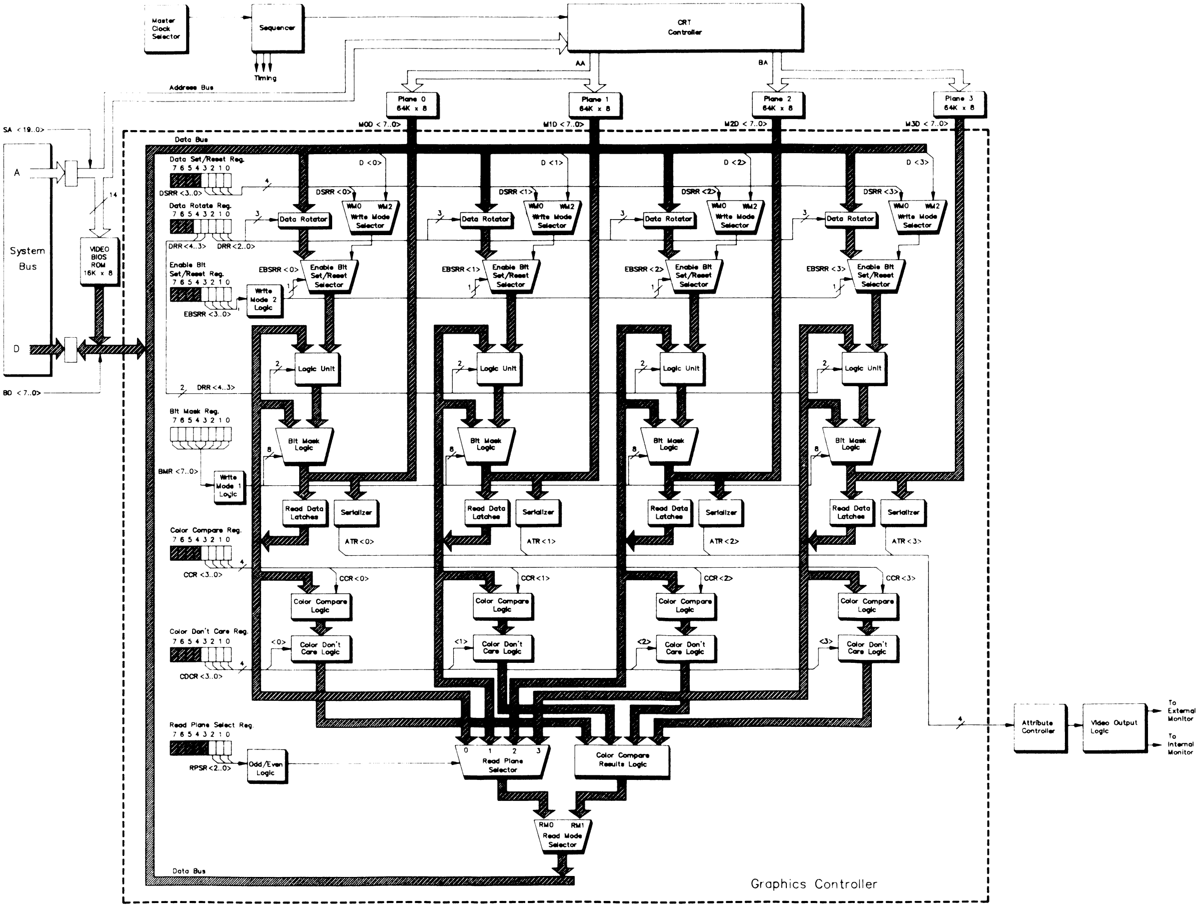
Write Mode 2 is used for selectively filling an area of the screen with a particular color. Each bit in the data byte from the system is expanded to 8 bits in each plane. Each bit loads 1 byte in each of the four planes.

Read Mode 0 is a normal Read mode. It allows one of the four planes to be read back to the system. Each byte is read directly to the system without processing.

Read Mode 1 provides hardware assistance in searching the screen for pixels of a given color.

Video BIOS uses only Write Mode 0 and Read Mode 0. If software needs to change modes, the Graphics Controller Mode register must be changed. (See Chapter 4, Section 4.4, Graphics Controller Registers.)

The following text describes Write Mode 0 and Read Mode 0 and further employs them to explain the fundamentals of the enhanced graphics operating mode. Read Mode 1 is then described and used to explain the Color Compare function of the enhanced graphics operating mode. Lastly, Write Modes 1 and 2 are briefly described. All of the modes are represented on the enhanced graphics block diagram, which is shown in Figure 2-15.


Master Clock Selector
Sequencer
Timing
CRT Controller
Address Bus
AA
BA
Plane 0 64K x 8
Plane 1 64K x 8
Plane 2 64K x 8
Plane 3 64K x 8
M0D < 7..0>
M1D < 7..0>
M2D < 7..0>
M3D < 7..0>
SA < 19..0>
Data Bus
System Bus
A
D
14
VIDEO BIOS ROM 16K x 8
BD < 7..0>
Data Set/Reset Reg.
7 6 5 4 3 2 1 0
DSRR <3..0>
4
D <0>
D <1>
D <2>
D <3>
DSRR <0>
DSRR <1>
DSRR <2>
DSRR <3>
WM0 WM2 Write Mode Selector
Data Rotate Reg.
7 6 5 4 3 2 1 0
3
Data Rotator
DRR <4..3>
DRR <2..0>
Enable Bit Set/Reset Reg.
7 6 5 4 3 2 1 0
EBSRR <3..0>
Write Mode 2 Logic
EBSRR < 0>
EBSRR < 1>
EBSRR < 2>
EBSRR < 3>
Enable Bit Set/Reset Selector
1
2
Logic Unit
2 DRR <4..3>
Bit Mask Reg.
7 6 5 4 3 2 1 0
BMR <7..0>
Write Mode 1 Logic
8
Bit Mask Logic
Read Data Latches
Serializer
ATR <0>
ATR <1>
ATR <2>
ATR <3>
Color Compare Reg.
7 6 5 4 3 2 1 0
CCR <3..0>
CCR <0>
CCR <1>
CCR <2>
CCR <3>
Color Compare Logic
Color Don't Care Reg.
7 6 5 4 3 2 1 0
CDCR <3..0>
<0>
<1>
<2>
<3>
Color Don't Care Logic
Read Plane Select Reg.
7 6 5 4 3 2 1 0
RPSR <2..0>
Odd/Even Logic
0 1 2 3 Read Plane Selector
Color Compare Results Logic
RM0 RM1 Read Mode Selector
Data Bus
Graphics Controller
Attribute Controller
Video Output Logic
To External Monitor
To Internal Monitor

Legend:
A = Address
D = Data
AA = Address for Planes 0,1
BA = Address for Planes 2,3

Figure 2-15. Block Diagram of the Enhanced Graphics Mode (BIOS Modes D, F, and 10)

Write Mode 0

To explain the enhanced graphics mode, assume that five dots in the sequence white, black, white, black, white are to be written to the upper-left corner of the CRT screen. A single byte then represents the codes for these five dots (5 bits). The remaining 3 bits in the byte are ignored.

Also assume that the:

- Video BIOS Mode 10 (640 x 350, enhanced color) is selected
- Screen memory address of the byte to be written is A0000h
- Background is gray
- Color code for the screen byte is:
  1 = White
  0 = Black

To start the Write process, the address of the byte that is about to be written appears on the address bus and is sent to all four planes by the CRT controller. The address is written so that, for each plane, the data byte to be written in the upper-left corner of the CRT screen is stored in the memory location that corresponds to that screen position.

Notice that in Figure 2-15 the graphics controller has been expanded from that shown in the previous descriptions of the text mode and the compatible graphics mode. The graphics controller consists of four identical circuits, one for each plane. The data byte to be written to memory appears on the data bus and is input to the four data rotators in the graphics controller.

The data byte from the system is processed simultaneously by each of the four circuits. The output of each circuit is stored in its respective video RAM plane. The serialized output from each plane is used as 1 bit in the 4-bit attribute code that is input to the attribute controller (which selects a color from the color palette). The attribute code input to the color pallette is shown below:

Attribute Code Input to Color Palette

Bits/Plane 3 2 1 0		
0 1 1 1	=	White Dot
0 0 0 0	=	Black Dot
1 0 0 0	=	Gray (Background)

When writing a white or black dot, the zero required in plane 3 is written to all bits of the byte. In plane 3, white dots are represented by zeros. However, in planes 2, 1, and 0, white dots are represented by ones. Black dots are represented by zeros in planes 3, 2, 1, and 0. Gray (background) dots are represented by ones in plane 3, whereas they are represented by zeros in planes 2, 1, and 0 .

The following paragraphs outline how plane 3 is forced to 0, for both white and black dots.

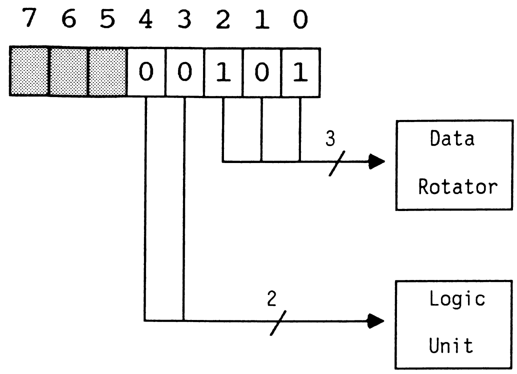
For ease of understanding, the ensuing text traces the data flow in the circuitry for plane 3. The other three circuits associated with planes 2, 1, and 0 function identically to that for plane 3, except that they may be processed differently. Processing is dependent on what was set in their respective control registers.

The end result is that the appropriate bit is output from each plane's serializer to form the attribute code input to the color palette.

Data Input to Rotator

The data byte is input to the data rotators. For ease of understanding, 05h has been loaded into the Data Rotate register, as shown in Figure 2-16.

Data Rotate Register


7 6 5 4 3 2 1 0
0 0 1 0 1
3
Data Rotator
2
Logic Unit

Bits 2, 1, 0	Number of Bits Rotated
000	0
001	1
010	2
011	3
100	4
101	5
110	6
111	7

Figure 2-16. Data Rotate Register

This loading causes the data rotator to move the bits in bit positions <4..0> in the data byte to bit positions <7..3> in the byte output from the rotator, as shown in Figure 2-17.

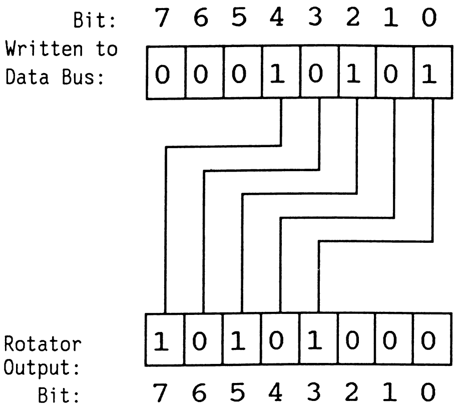
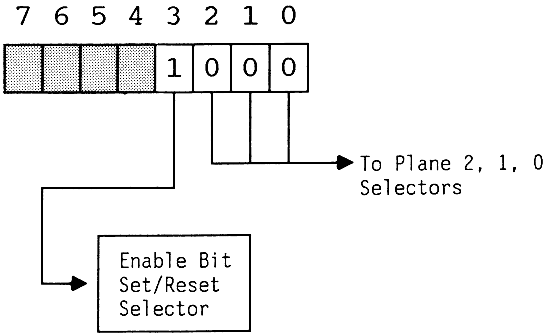

Bit: 7 6 5 4 3 2 1 0
Written to Data Bus:
0 0 0 1 0 1 0 1
Rotator Output:
1 0 1 0 1 0 0 0
Bit: 7 6 5 4 3 2 1 0

Figure 2-17. Data Rotation

To force plane 3 to 0, a 1 is set in bit <3> of the Enable Bit Set/Reset register, as shown in Figure 2-18.

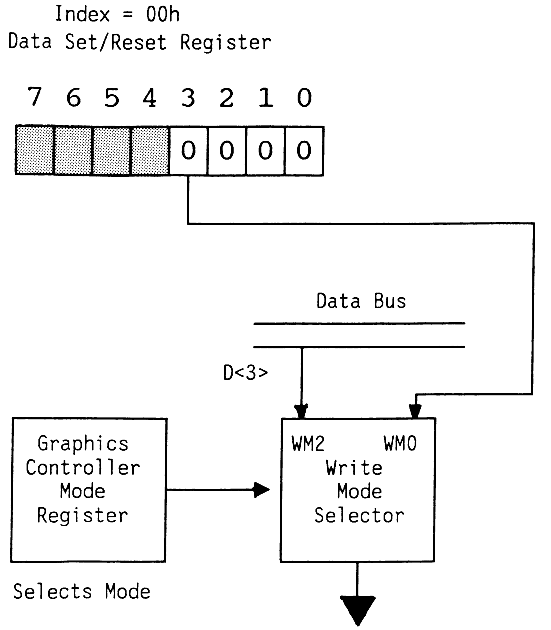
Index = 01h
Enable Bit Set/Reset Register


7 6 5 4 3 2 1 0
1 0 0 0
To Plane 2, 1, 0 Selectors
Enable Bit Set/Reset Selector

0 = Use Data Rotator Output
1 = Use Write Mode Select Data

Figure 2-18. Enable Bit Set/Reset Register

Because the Write mode selector is set to Write Mode 0 by the Graphics Controller Mode register, as shown in Figure 2-19, the Write Mode 0 (WM0) input gets a 0 from the Data Set/Reset register. This 0 is fed through the Write Mode selector to the Enable Bit Set/Reset selector. Zeros are placed in all eight bit positions of the input to the Enable Bit Set/Reset selector.


Index = 00h
Data Set/Reset Register
7 6 5 4 3 2 1 0
0 0 0 0
Data Bus
D<3>
Graphics Controller Mode Register
WM2 WM0
Write Mode Selector
Selects Mode

0 = Supply zeros to all 8 bits of that plane's byte
1 = Supply ones

Figure 2-19. Write Mode Selector

The eight zeros are output from the Enable Bit Set/Reset register to the logic unit. The logic unit can do one of four things:

1. Replace the contents of screen memory with incoming data from the Enable Bit Set/Reset selector.

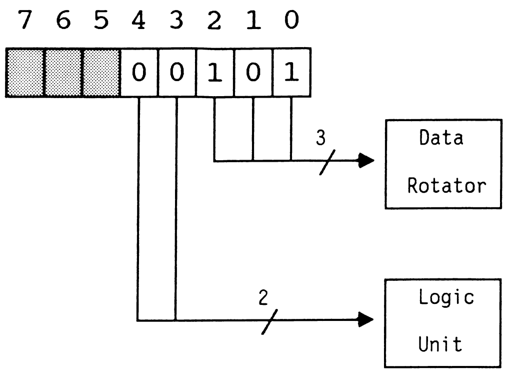
2. Perform a logic "AND" operation between the existing contents of screen memory, as held in the Read Data latch, and the incoming data from the Enable Bit Set/Reset selector.

3. Perform a logic "OR" operation with data as noted in number 2 above.

4. Perform a logic "XOR" (exclusive-or) operation with data as noted in number 2 above.

The logic unit receives data from the Enable Bit Set/Reset selector and from the Read Data latch. The logic unit performs its operations as directed by bits <4..3> of the Data Rotate register, which is shown in Figure 2-20.

Data Rotate Register



Bits 4	Bits 3	Logic Operation	
0	0	REPLACE	No merge. Screen data are replaced with incoming bits.
0	1	AND	Bits merged with data currently on screen (in Read Data latch).
1	0	OR	
1	1	XOR	

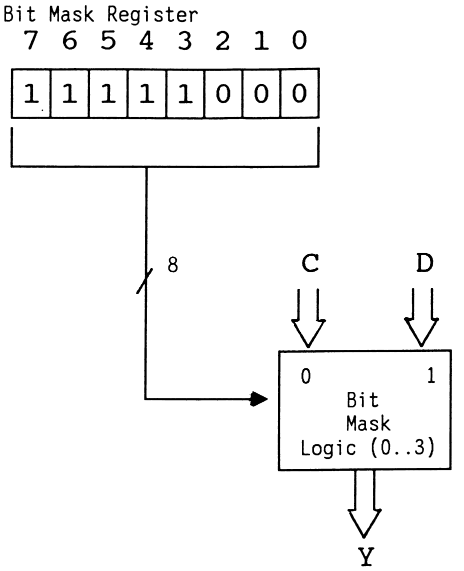
Figure 2-20. Data Rotate Register

Note: The Data Rotate register controls all four data rotators and all four logic units. Similarly, the Bit Mask register controls all four bit masks. Therefore, when bits <4..3> in the Data Rotate register are set to zeros, all four data rotators perform a REPLACE function.

With zeros in bit positions <4..3> of the Data Rotate register, the logic unit does a REPLACE function for all four planes. For plane 3, the logic unit receives the screen data from the Read Data latch replaced with all zeros. All zeros are output to the Bit Mask selector.

The Bit Mask logic gets input from the data latch and from the logic unit. The Bit Mask logic allows existing bits in the screen memory byte (as held in the data latch) to be selectively replaced, as specified by the contents of the Bit Mask register.

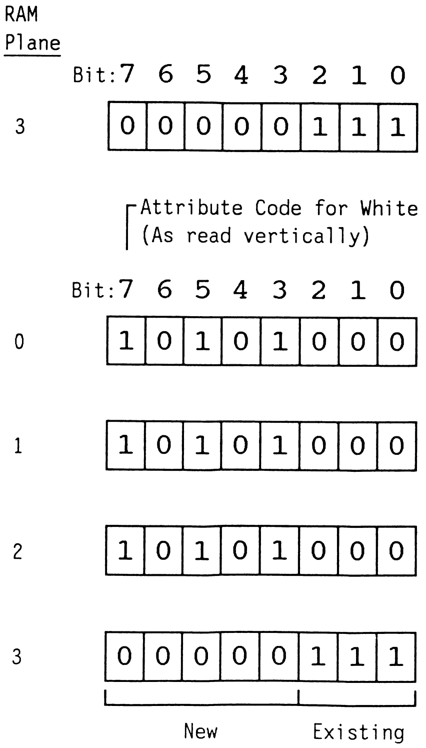
In this case, the Bit Mask register, shown in Figure 2-21, is loaded with a mask that blocks screen memory bits 2, 1, and 0. Writing 5 bits only in the first five places means that the last 3 bits are masked. The 3 bits as represented in the Read Data latch are preserved. Then new bits are written in the first five places. Even though 8 bits at a time are written to the screen, fewer than 8 bits can be written by using the mask.



Bit
Cell
7..0 0 = Preserve existing screen memory bit
1 = Replace with incoming data bit

Figure 2-21. Bit Mask Register

The output of the bit mask logic is shown in Figure 2-22. The top row of the figure shows the bit pattern stored in plane 3. The bottom four rows show the output of all bit mask logic, including plane 3.



RAM Plane	Bit:7	6	5	4	3	2	1	0
3	0	0	0	0	0	1	1	1

Attribute Code for White
(As read vertically)

	Bit:7	6	5	4	3	2	1	0
0	1	0	1	0	1	0	0	0
1	1	0	1	0	1	0	0	0
2	1	0	1	0	1	0	0	0
3	0	0	0	0	0	1	1	1

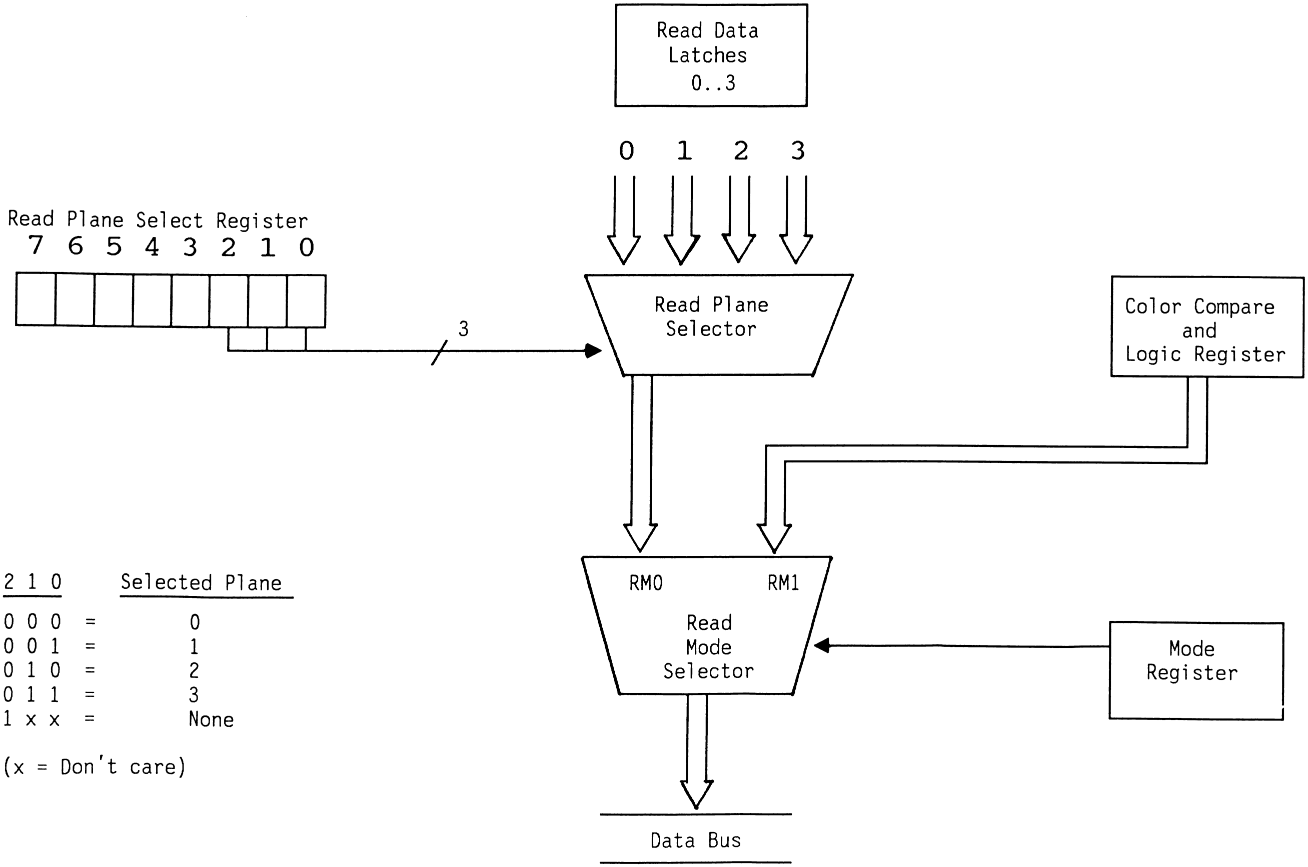
Figure 2-22. Bit Mask Logic Output

The bit patterns stored in planes 0 through 3 go to serializers 0 through 3. There, the serializers shift the data out to the left. Each serializer forms an input to the attribute controller. In this case, the first bit shifted out of each serializer (bit <7>) forms the attribute code that selects the color white (0111) from the color palette.

The next 4 bits that are shifted out form the attribute code for black (0000). The left-shift continues until all dots are written to the screen.

Read Mode 0

Read Mode 0 allows any one of the planes to be read. Figure 2-23 illustrates the process. Read Mode 0 is selected by the Graphics Controller Mode register (see Chapter 4, Section 4.4, Graphics Controller Registers). The plane to be read is selected by entering the appropriate code in the Read Plane Select register. When a Read cycle occurs, all four planes are read and their data latched, but only the data from the designated plane is used. To read all four individual planes, this sequence must occur four times.


Read Data Latches 0..3
0
1
2
3
Read Plane Select Register
7 6 5 4 3 2 1 0
3
Read Plane Selector
Color Compare and Logic Register
RM0
RM1
Read Mode Selector
Mode Register
Data Bus

2 1 0		Selected Plane
0 0 0	=	0
0 0 1	=	1
0 1 0	=	2
0 1 1	=	3
1 x x	=	None

(x = Don't care)

Figure 2-23. Read Mode 0 Data Path

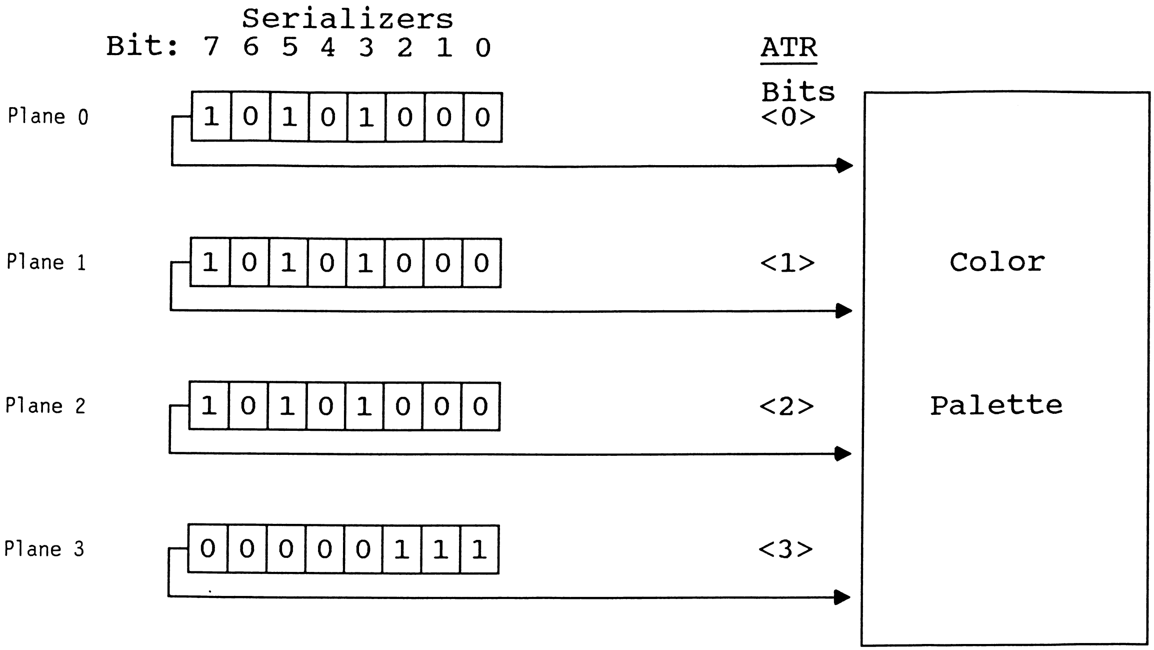
System Read cycle is initiated:

1. A0000h (address of first byte in screen memory) is sent through the CRT controller to all four planes.
2. Data are read from all four planes (32 bits) and arrive at the input of the Read Data latches. The latches hold data for use during Merge and Mask operations, previously described.
3. Data is then input to the Read Plane selector from the four Read Data latches.
4. Data from one of the planes are returned by the Read Mode selector to the data bus.
5. Data are returned to the specified system registers for any manipulation that may be required.

Display Cycle

The display cycle moves data from the planes to the CRT. Figure 2-24 illustrates the following process.

1. The address counter in the CRT controller supplies a 16-bit address to all four planes of the byte to be moved to the screen.
2. Under control of the sequencer, data are read from the four planes into the four serializers.
3. The leftmost bit <7> is shifted out of all four serializers simultaneously. Bit <7> from each serializer forms the 4-bit attribute code for selection of a color from the color palette, with ATR bit 0 = plane 0.


Serializers
Bit: 7 6 5 4 3 2 1 0
ATR
Bits
Plane 0
1 0 1 0 1 0 0 0
<0>
Plane 1
1 0 1 0 1 0 0 0
<1>
Plane 2
1 0 1 0 1 0 0 0
<2>
Plane 3
0 0 0 0 0 1 1 1
<3>
Color
Palette

ATR Bits 3 2 1 0	Codes
0 1 1 1	White Dot
0 0 0 0	Black Dot
1 0 0 0	Gray Dot

Figure 2-24. Data Display Cycle

Read Mode 1

Read Mode 1 allows software to determine which pixels match a given color. This mode allows screen memory to be rapidly scanned to find pixels that match a given color. The search for white pixels in the alternating pattern described in Write Mode 0 is outlined below and is illustrated in Figure 2-25 and Figure 2-26.

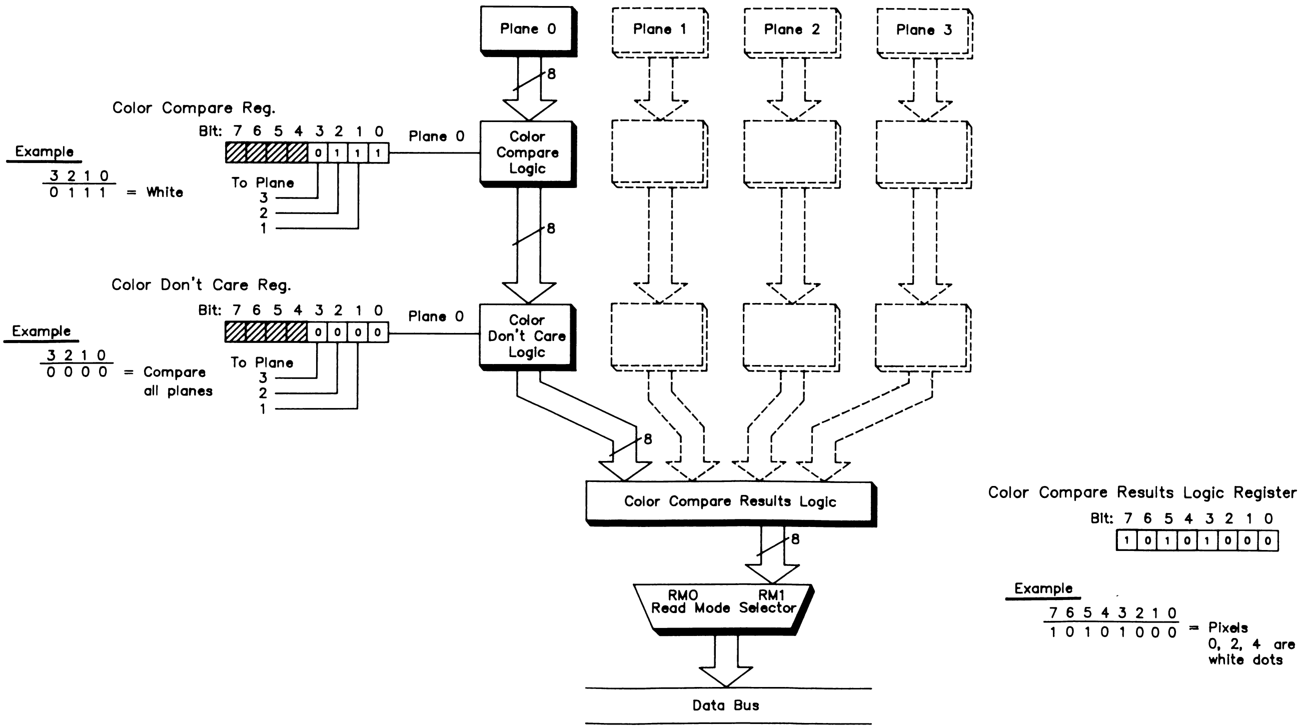

Plane 0
Plane 1
Plane 2
Plane 3
8
Color Compare Reg.
Bit: 7 6 5 4 3 2 1 0
0 1 1 1
Plane 0
Color Compare Logic
Example
3 2 1 0
0 1 1 1 = White
To Plane
3
2
1
8
Color Don't Care Reg.
Bit: 7 6 5 4 3 2 1 0
0 0 0 0
Plane 0
Color Don't Care Logic
Example
3 2 1 0
0 0 0 0 = Compare all planes
To Plane
3
2
1
8
Color Compare Results Logic
8
RM0
RM1
Read Mode Selector
Data Bus
Color Compare Results Logic Register
Bit: 7 6 5 4 3 2 1 0
1 0 1 0 1 0 0 0
Example
7 6 5 4 3 2 1 0
1 0 1 0 1 0 0 0 = Pixels 0, 2, 4 are white dots

Figure 2-25. Color Comparison

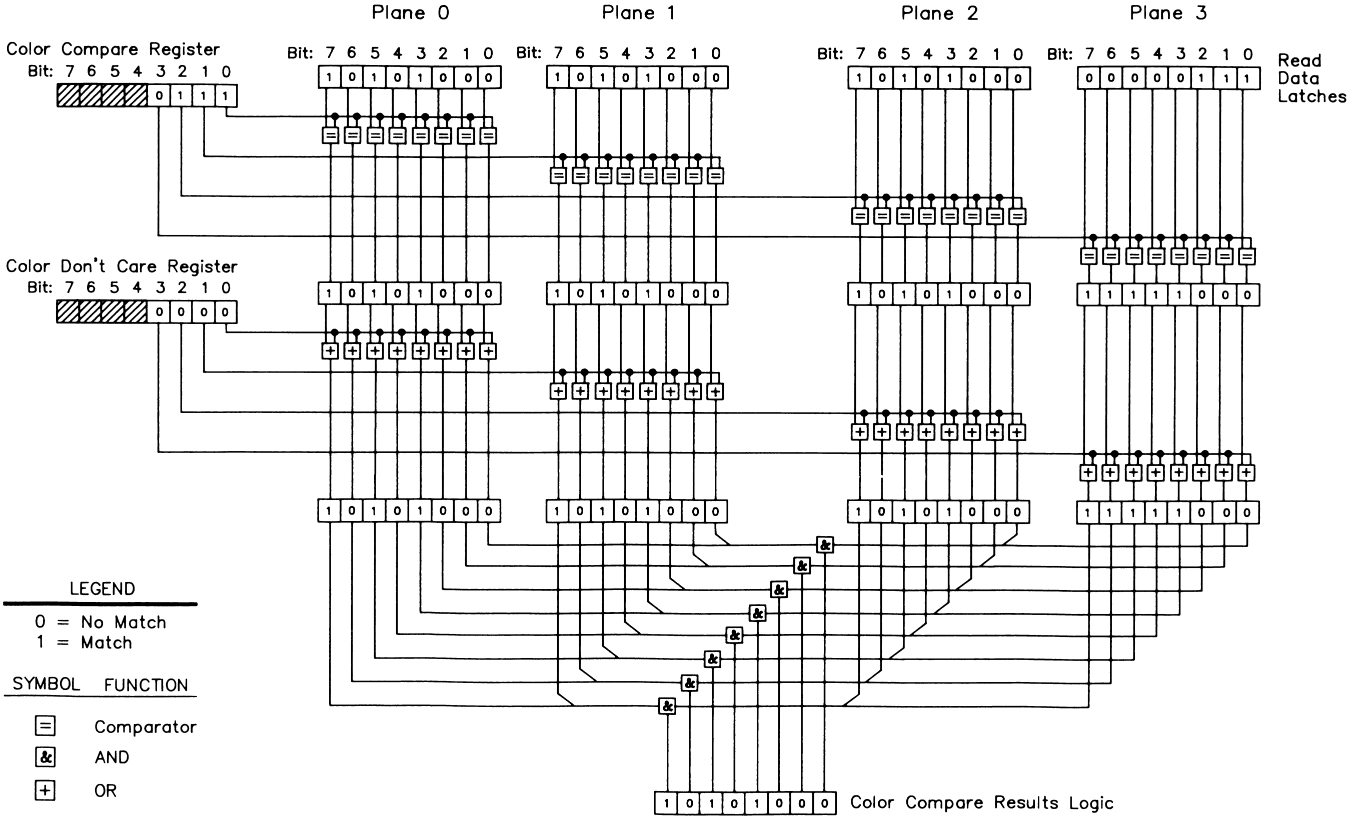

Plane 0
Plane 1
Plane 2
Plane 3
Color Compare Register
Bit: 7 6 5 4 3 2 1 0
0 1 1 1
Color Don't Care Register
Bit: 7 6 5 4 3 2 1 0
0 0 0 0
Bit: 7 6 5 4 3 2 1 0
1 0 1 0 1 0 0 0
0 0 0 0 0 1 1 1
1 1 1 1 1 0 0 0
Read Data Latches
Color Compare Results Logic
1 0 1 0 1 0 0 0
LEGEND
0 = No Match
1 = Match
SYMBOL FUNCTION
= Comparator
& AND
+ OR

Figure 2-26. Results of Color Comparison

The following steps comprise the setup process for Read Mode 1.

Setup:

1. The CRT controller is set to Read Mode 1 via the Graphics Controller Mode register.

2. The color code to be matched is loaded into the Graphics Controller Color Compare register.

3. The planes to be ignored are determined by bits set in the Graphics Controller Color Don't Care register.

The following steps are representative of the operation for Read Mode 1.

Operation:

1. A0000h (address of the first byte in screen memory) is sent through the CRT controller to all four planes in parallel.

2. Data are read in parallel from all four planes (32 bits) and arrives at the input of the Read Data latches. (These latches also hold data for use during the Write Mode.)

3. Data from each Read Data latch is input to the Graphics Controller Color Compare logic.

4. Each of the 8 incoming bits are compared to the appropriate attribute bit from the Color Compare register. Bits <3..0> in the Color Compare register represent the attribute code for a particular color. Bit <0> of the Color Compare register is compared with the 8 bits from the Read Data latch for plane 0. The remaining 3 bits are compared similarly: bits 1, 2, and 3 of the Color Compare register corresponding to planes 1, 2, and 3 Read Data latches.

   The bits coming into the Color Compare register yield either ones on the corresponding output when a match occurs or a zero when there is no match. Refer back to Column 2 of Figure 2-26, which illustrates no matches occurring.

5. The output from the Color Compare logic goes to the Color Don't Care logic. Since zeros are in the Color Don't Care register, the output of the Color Compare logic is passed through.

   Note that in monochrome (BIOS Mode F) the Color Don't Care register would have 1010 loaded. This loading would occur because only planes 0 and 2 would be used. Therefore, the outputs from planes 1 and 3 would be excluded.

6. The output from the Color Don't Care logic goes to the Color Compare Results logic. When all planes have corresponding bits matched, bit <0> in the result is a 1 (meaning a match) when all four bit <0> outputs of the Color Don't Care logic are a 1. Bits <7..1> of the result logic are generated in the same manner.

7. The byte output from the Color Compare Results logic is passed to the Read Mode selector, and then to the data bus.

   Similarly, planes 1, 2, and 3 relate to bits 1, 2, and 3. For example, writing a data byte of 00000100 (binary) to video memory in the specified address location fills planes 0, 1, and 3 with all zeros and plane 1 with all ones. Bit <0> of the Color Don't Care register determines whether the input should be passed through unchanged or whether that plane should be excluded from the comparison by outputting all ones.

Write Mode 1

In Write Mode 1, the data written to memory are simply the contents of the Read Data Latches. The data rotator, Enable Set/Reset logic, logic unit, and bit mask are completely bypassed. Writes to the individual planes can be disabled by altering the Write Plane Mask register, located in the sequencer.

Write Mode 2

In Write Mode 2, plane 0 at the addressed location is filled with the contents of bit <0> of the data written. The AND/OR/XOR functions and the bit mask operate as they did for Write Mode 0. The data rotator and Enable Set/Reset logic are bypassed.

Dual-Mode Support

The COMPAQ Dual-Mode Monitor directly accepts the RGBI video signals output for all 200-scan line formats. However, the video output for all 350-scan line formats requires conversion when a COMPAQ Dual-Mode Monitor is being driven. The 350-scan line formats have two categories of video output signals: RrGgBb for enhanced color and VI (video and intensity) for monochrome.

The RrGgBb signals that are output for the 350-scan line color format represent codes for 64 colors. The video output logic maps these 64 color codes to 16 RGBI codes when a COMPAQ Dual-Mode Monitor is being driven. Table 2-9 gives the signal conversions.

Table 2-9. Conversion of RrGgBb Colors to RGBI Colors

RrGgBb Palette Entry Bits											
5	4	3	2	1	0	Color	RGBI Output				DM Display Relative
r	g	b	R	G	B	Code	I	R	G	B	Intensity
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	1	1	0	0	0	1	1
0	0	0	0	1	0	2	0	0	1	0	2
0	0	0	0	1	1	3	0	0	1	1	3
0	0	0	1	0	0	4	0	1	0	0	5
0	0	0	1	0	1	5	0	1	0	1	6
0	0	0	1	1	0	6	0	1	1	0	9
0	0	0	1	1	1	7	0	1	1	1	10
0	0	1	0	0	0	8	0	0	0	1	1
0	0	1	0	0	1	9	1	0	0	1	4
0	0	1	0	1	0	10	0	0	1	1	3
0	0	1	0	1	1	11	1	0	1	1	8
0	0	1	1	0	0	12	0	1	0	1	6
0	0	1	1	0	1	13	1	1	0	1	12
0	0	1	1	1	0	14	0	1	1	1	10
0	0	1	1	1	1	15	1	0	0	1	4
0	1	0	0	0	0	16	0	0	1	0	2
0	1	0	0	0	1	17	0	0	1	1	3
0	1	0	0	1	0	18	1	0	1	0	7
0	1	0	0	1	1	19	1	0	1	1	8
0	1	0	1	0	0	20	0	1	1	0	9
0	1	0	1	0	1	21	0	1	1	1	10
0	1	0	1	1	0	22	1	1	1	0	13
0	1	0	1	1	1	23	1	0	1	0	7

Legend: DM = COMPAQ Dual-Mode Monitor

(Continued)

Table 2-9. Conversion of RrGgBb Colors to RGBI Colors (Continued)

RrGgBb Palette Entry Bits											DM Display
5	4	3	2	1	0	Color	RGBI Output				Relative
r	g	b	R	G	B	Code	I	R	G	B	Intensity
0	1	1	0	0	0	24	0	0	1	1	3
0	1	1	0	0	1	25	1	0	1	1	8
0	1	1	0	1	0	26	1	0	1	1	8
0	1	1	0	1	1	27	1	0	1	1	8
0	1	1	1	0	0	28	0	1	1	1	10
0	1	1	1	0	1	29	1	1	1	1	14
0	1	1	1	1	0	30	1	1	1	1	14
0	1	1	1	1	1	31	1	0	1	1	8
1	0	0	0	0	0	32	0	1	0	0	5
1	0	0	0	0	1	33	0	1	0	1	6
1	0	0	0	1	0	34	0	1	1	0	9
1	0	0	0	1	1	35	0	1	1	1	10
1	0	0	1	0	0	36	1	1	0	0	11
1	0	0	1	0	1	37	1	1	0	1	12
1	0	0	1	1	0	38	1	1	1	0	13
1	0	0	1	1	1	39	1	1	0	0	11
1	0	1	0	0	0	40	0	1	0	1	6
1	0	1	0	0	1	41	1	1	0	1	12
1	0	1	0	1	0	42	0	1	1	1	10
1	0	1	0	1	1	43	1	1	1	1	14
1	0	1	1	0	0	44	1	1	0	1	12
1	0	1	1	0	1	45	1	1	0	1	12
1	0	1	1	1	0	46	1	1	1	1	14
1	0	1	1	1	1	47	1	1	0	1	12

(Continued)

Table 2-9. Conversion of RrGgBb Colors to RGBI Colors (Concluded)

RrGgBb Palette Entry Bits											DM Display
5	4	3	2	1	0	Color	RGBI Output				Relative
r	g	b	R	G	B	Code	I	R	G	B	Intensity
1	1	0	0	0	0	48	0	1	1	0	9
1	1	0	0	0	1	49	0	1	1	1	10
1	1	0	0	1	0	50	1	1	1	0	13
1	1	0	0	1	1	51	1	1	1	1	14
1	1	0	1	0	0	52	1	1	1	0	13
1	1	0	1	0	1	53	1	1	1	1	14
1	1	0	1	1	0	54	1	1	1	0	13
1	1	0	1	1	1	55	1	1	1	0	13
1	1	1	0	0	0	56	1	0	0	0	0
1	1	1	0	0	1	57	1	0	0	1	4
1	1	1	0	1	0	58	1	0	1	0	7
1	1	1	0	1	1	59	1	0	1	1	8
1	1	1	1	0	0	60	1	1	0	0	11
1	1	1	1	0	1	61	1	1	0	1	12
1	1	1	1	1	0	62	1	1	1	0	13
1	1	1	1	1	1	63	1	1	1	1	14

Conversion of the VI signal to an RGBI signal is necessary when a COMPAQ Dual-Mode Monitor is displaying a monochrome format (BIOS Modes 7 and F). Relative intensities shown for the COMPAQ Dual-Mode Monitor in Tables 2-9 and 2-10 are on a scale from 0 (black) to 14 (highlighted), with 10 being the default. This conversion is given in Table 2-10.

Table 2-10. Conversion of VI Signals to RGBI Intensities

VI Palette Entry Bits											DM Display
5	4	3	2	1	0	Color	RGBI Output				Relative
-	I	V	-	-	-	Code	I	R	G	B	Intensity
X	0	0	X	X	X	0	0	0	0	0	0
X	0	1	X	X	X	8	0	1	1	1	10
X	1	0	X	X	X	16	1	0	0	0	0
X	1	1	X	X	X	24	1	1	1	1	14

Legend: DM = COMPAQ Dual-Mode Monitor
X = Ignored

LIGHTPEN

The lightpen can determine the address of a character on the CRT screen. This allows the user to select, for example, items from a menu. The lightpen is also used in some computer-aided design (CAD) applications.

A screen format of 80 columns by 25 rows provides 2000 available character locations, each with a separate video memory address. If the lightpen is placed over any of these character locations, the character address that is read is synchronized with the light emanating from any scan line of that character that the lightpen detects (usually the scan line at the top of the character).

If the character size is 8 x 14, each of the 14 scan lines comprising the character signifies the same video memory address. The character location in the upper-left corner of the CRT screen has a video memory (plane 0) address of 0000h. If the letter "A" were put in that location, each of the 14 scan lines required to produce the letter "A" would have a character (plane 0) address of 0000h, and that is the address that ultimately would be transmitted back to the system.

Operation

When reading the following operating theory, refer to Figure 2-27.

Operation:

1. The lightpen is placed over a character on the screen.
2. The user then presses switch LPSW on the lightpen.
3. Bit <2> of the Input Status 1 register is set to 0.
4. Software reads the register and detects a change in bit <2>.
5. Software clears the lightpen latch by writing any value to address 3DBh (Lightpen Latch Reset register). The LPCLR- line of the lightpen latch goes briefly to 0, resetting it.

6. When the character covered by the lightpen is scanned by the first scan line, the light from that scan line is sensed by the lightpen photodiode.

7. After signal amplification and shaping, the output of the lightpen is an active-low (0) TTL signal.

8. This transition to 0 is inverted and input to the lightpen latch, which is then set.

9. The software reads a 1 in bit <1> of the Input Status 1 register.

10. At the same time the 1 is read, the CRT controller address counter is generating the address of the character covered by the lightpen. This address appears at the input of the lightpen register.

11. At the same instant that a 1 appears in bit <1> of the Input Status 1 register, the LPENSTB signal causes the lightpen register to store the current address from the address counter (a 16-bit value).

12. Software then reads the lightpen register, which now contains the address of the character read by the lightpen.

13. Software corrects for lightpen response delays.

14. The address is then converted to the row and column designation via the lightpen software.

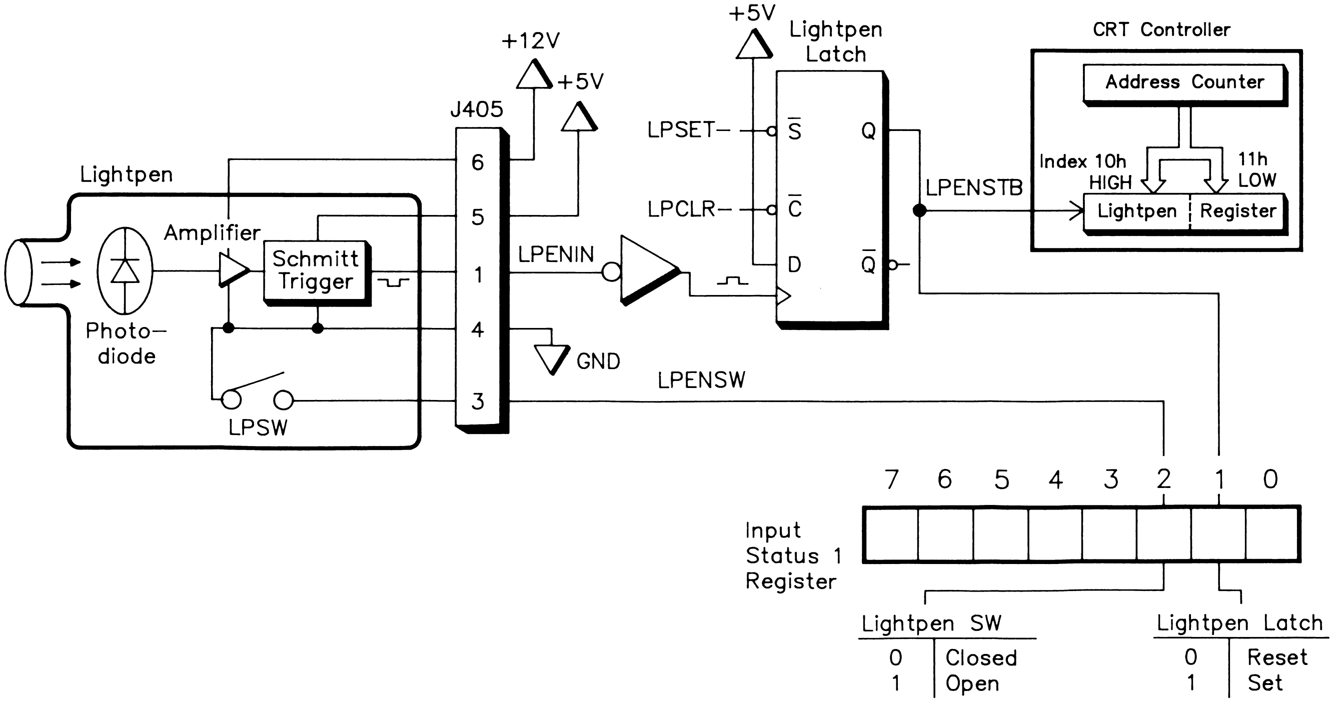

+12V
+5V
J405
Lightpen
Amplifier
Schmitt Trigger
Photo–diode
LPSW
6
5
1
4
3
LPENIN
GND
LPENSW
+5V
Lightpen Latch
LPSET–
LPCLR–
$\overline{S}$
$\overline{C}$
D
Q
$\overline{Q}$
LPENSTB
CRT Controller
Address Counter
Index 10h HIGH
11h LOW
Lightpen
Register
7 6 5 4 3 2 1 0
Input Status 1 Register
Lightpen SW
0 Closed
1 Open
Lightpen Latch
0 Reset
1 Set

Figure 2-27. Lightpen Functional Block Diagram

2.3 POWER-ON SELF-TEST CODES

The COMPAQ Enhanced Color Graphics Board power-on self-test codes are represented by two-digit hexadecimal numbers that are output to port 084h. The first (left) digit specifies the group or class, while the second (right) digit indicates more specific information. Class 0 is a special group that does not indicate an error. Instead, it indicates which section or test is under way during the self-test. All classes other than Class 0 indicate power-on self-test errors.

Table 2-11 is a complete list of the COMPAQ Enhanced Color Graphics Board BIOS ROM self-test codes and their explanations.

Table 2-11. BIOS ROM Power-On Self-Test Codes and Their Explanations

Code	Explanation	Cause
Class 0 - Self-test process status indication:		
00	Entry into the video option ROM	
01	COMPAQ-Specific Registers test	
02	Vertical Sync test	
03	Horizontal Sync test	
04	[Reserved]	
.		
.		
.		
07	[Reserved]	
08	Video Bit Off test	
09	Video Bit On test	
0A	Video Memory test	
0B	[Reserved]	
.		
.		
.		
0F	[Reserved]	

(Continued)

Table 2-11. BIOS ROM Power-On Self-Test Codes and Their Explanations (Continued)

Code	Explanation	Cause
Class 1 - COMPAQ-Specific Registers Test:		
10	No Master Mode register	Read/Write test failed
11	No video modes supported	Environment register all ones
12	No initial (power-on) mode	Invalid initial register value
13	Initial mode not supported	(See Note 1)
14	No monitor present	Display register all zeros
15	Monitor does not support mode	(See Note 2)
16	[Reserved]	
.		
.		
.		
1F	[Reserved]	

(Continued)

Notes:

1. The initial or power-on mode selected by configuration switch SW1 settings is NOT supported by this video adapter. The probable cause for this is incorrectly set switches.

2. The monitor selected does not support the selected initial or power-on mode. The COMPAQ Enhanced Color Graphics Board will initialize, but video will probably be absent. The probable cause for this is incorrectly set switches.

(Continued)

Table 2-11. BIOS ROM Power-On Self-Test Codes and Their Explanations (Continued)

Code	Explanation	Cause
Class 2 - Vertical Sync Tests:		
20	No vertical sync present	No sync found in 50 ms.
21	Vertical sync invalid	(See Note 3)
22	[Reserved]	
.		
.		
.		
2F	[Reserved]	

Notes:

3. The vertical sync period is timed with the system counter-timer to ensure that the period is within 5 percent of the correct value. The timer uses a 1.193-MHz clock rate to achieve approximately a 1-microsecond resolution. Limiting values are based on the monitor type and are given below:

	50-Hz Monitor	60-Hz Monitor
Limiting Values	47.5 Hz to 52.5 Hz	57.0 Hz to 63.0 Hz

(Continued)

Table 2-11. BIOS ROM Power-On Self-Test Codes and Their Explanations (Continued)

Code	Explanation	Cause
Class 3 - Horizontal Sync Tests:		
30	No horizontal sync present	No sync found in 20 ms.
31	Horizontal sync invalid	(See Note 4)
32	[Reserved]	
.		
.		
.		
7F	[Reserved]	

Notes:

4. The number of horizontal sync periods in one vertical frame is counted to ensure that the number is within +/- 5 scan lines of the correct value. The limiting values are based on the monitor type and mode and are given below:

	200-Scan Line Mode	350-Scan Line Mode
Limiting Values	195..205	345..355

(Continued)

Table 2-11. BIOS ROM Power-On Self-Test Codes and Their Explanations (Continued)

Code	Explanation	Cause
Class 8 - Video Output OFF Tests:		
80	[Reserved]	
81	Primary blue (B) stuck ON (See Note 5)	U1 fault (82C433)
82	Primary red (R) stuck ON	U1 fault (82C433)
83	Primary red (R) and primary blue (B) stuck ON	U1 fault (82C433)
84	[Reserved]	
85	Secondary green (g) stuck ON	U1 fault (82C433)
86	Secondary red (r) stuck ON	U1 fault (82C433)
87	Secondary red (r) and secondary green (g) stuck ON	U1 fault (82C433)
88	[Reserved]	
89	Primary green (G) stuck ON	U1 fault (82C433)
8A	Secondary blue (b) stuck ON	U1 fault (82C433)
8B	Secondary blue (b) and primary green (G) stuck ON	U1 fault (82C433)
8C	[Reserved]	
.		
.		
.		
8F	[Reserved]	

(Continued)

Notes:

5. One of the six video outputs was found to be stuck in the ON condition. Video outputs are read back in three groups of two colors each.

(Continued)

Table 2-11. BIOS ROM Power-On Self-Test Codes and Their Explanations (Continued)

Code	Explanation	Cause
Class 9 - Video Output ON Tests:		(See Note 6)
90	Primary red (R) and primary blue stuck OFF	U1 fault (82C433)
91	Primary red (R) stuck OFF	U1 fault (82C433)
92	Primary blue (B) stuck OFF	U1 fault (82C433)
93	[Reserved]	
94	Secondary red (r) and secondary green (g) stuck OFF	U1 fault (82C433)
95	Secondary red (r) stuck OFF	U1 fault (82C433)
96	Secondary green (g) stuck OFF	U1 fault (82C433)
97	[Reserved]	
98	Secondary blue (b) and primary green (G) stuck OFF	U1 fault (82C433)
99	Secondary blue (b) stuck OFF	U1 fault (82C433)
9A	Primary green (G) stuck OFF	U1 fault (82C433)
9B	[Reserved]	
.		
.		
.		
9F	[Reserved]	

Notes:

6. One of the six video outputs was found to be stuck in the OFF condition. Video outputs are read back in three groups of two colors each.

(Continued)

Table 2-11. BIOS ROM Power-On Self-Test Codes and Their Explanations (Concluded)

Code	Explanation	Cause
Class A - Video Memory Tests:		
A0	Video memory failure	U5 fault (4164-15)
A1	Video memory failure	U11 fault (4164-15)
A2	[Reserved]	
A3	[Reserved]	
A4	Video memory failure	U6 fault (4164-15)
A5	Video memory failure	U12 fault (4164-15)
A6	[Reserved]	
A7	[Reserved]	
A8	Video memory failure	U7 fault (4164-15)
A9	Video memory failure	U13 fault (4164-15)
AA	[Reserved]	
AB	[Reserved]	
AC	Video memory failure	U8 fault (4164-15)
AD	Video memory failure	U14 fault (4164-15)
AE	[Reserved]	
AF	Video memory soft error	(See Note 7)
B0	[Reserved]	
.		
.		
.		
FF	[Reserved]	

Notes:

7. If, during the attempt to isolate the video memory error to a specific chip, the error does not occur again, the error was a "soft" error instead of a "hard" failure. Soft errors indicate that a memory chip is intermittent.

Figure 2-28 shows the location, integrated chip designator, and corresponding error code for the video RAM planes.

	Chip Column 0	Error Code	Chip Column 1	Error Code
Chip Row 0	U5 Plane 1 Bits <7..4>	A0	U11 Plane 3 Bits <7..4>	A1
Chip Row 1	U6 Plane 0 Bits <7..4>	A4	U12 Plane 2 Bits <7..4>	A5
Chip Row 2	U7 Plane 1 Bits <3..0>	A8	U13 Plane 3 Bits <3..0>	A9
Chip Row 3	U8 Plane 0 Bits <3..0>	AC	U14 Plane 2 Bits <3..0>	AD

Figure 2-28. Video Memory Chip Layout and Location Codes

TABLE OF CONTENTS

CHAPTER 3
SPECIFICATIONS

3.1 HARDWARE 3-1
3.2 SOFTWARE 3-1

Chapter 3
SPECIFICATIONS

This chapter provides the hardware and software specifications for the COMPAQ Enhanced Color Graphics Board.

3.1 HARDWARE

This section gives the dimensions of the COMPAQ Enhanced Color Graphics Board and its power requirements.

Board Dimensions: 4.2 inches x 13.3 inches
(107 mm x 338 mm)

Power Consumption:

DC Voltage	Current (Amperes)
+5V	2.0
+12V	0.2

Table 3-1 lists the board connectors. (Refer to Chapter 6, Section 6.3 entitled "Connectors", for a detailed description of these connectors, their locations, and their interface signals.)

Table 3-1. Board Connectors

Connector	Location	Type
External Monitor	J404	9-pin subminiature D
Internal Monitor	J401	12-pin header
Special Features	J4	32-pin header
RCA	J1, J2	Dual RCA connector
Lightpen	J405	6-pin header

3.2 SOFTWARE

This section provides software specifications for proper operation of the board. It lists the software registers, BIOS mode operating characteristics, I/O addresses, and video BIOS calls.

Tables 3-2, 3-3, 3-4, 3-5, and 3-6 list the software registers and their indices (00h, 01h, and so on) for the board.

Table 3-2. Control and Status Registers

Address	Register	Read/Write
3BAh	Input Status 1 (Monochrome)	R
3BAh	Feature Control (Monochrome)	W
3C2h	Input Status 0	R
33C2h	Miscellaneous Output	W
3C6h (Note)	Control Mode	R/W
3DAh	Input Status 1 (Color)	R
3DAh	Feature Control (Color)	W
3DBh	Lightpen Latch Reset (Color)	W
3DCh	Lightpen Latch Set (Color)	W
7C6h (Note)	Environment	R
BC6h (Note)	Display Type	R
FC6h (Note)	Initial Mode	R

Note: These COMPAQ-specific registers decode all 16 bits in I/O space; therefore, bit <15..12> must equal 0.

Table 3-3. Sequencer Registers

Address	Register	Read/Write
3C4h	Index	W
3C5h	Data	W
00h	Reset	W
01h	Clocking Mode	W
02h	Write Plane Mask	W
03h	Character Font Select	W
04h	Memory Mode	W

Table 3-4. Graphics Controller Registers

Address	Register	Read/Write
3CAh	Graphics Controller 2 Position	W
3CCh	Graphics Controller 1 Position	W
3CEh	Index	W
3CFh	Data	W
00h	Data Set/Reset	W
01h	Enable Bit Set/Reset	W
02h	Color Compare	W
03h	Data Rotate	W
04h	Read Map Select	W
05h	Mode	W
06h	Miscellaneous	W
07h	Color Don't Care	W
08h	Bit Mask	W

Table 3-5. CRT Controller Registers

Address	Register	Read/Write
3B4h	CRT Controller Index (Monochrome)	W
3B5h (See Note)	CRT Controller Data (Monochrome)	R/W
3D4h	CRT Controller Index (Color)	W
3D5h (See Note)	CRT Controller Data (Color)	R/W
00h	Horizontal Total	W
01h	Horizontal Display End	W
02h	Start Horizontal Blank	W
03h	End Horizontal Blank	W
04h	Start Horizontal Sync	W
05h	End Horizontal Sync	W
06h	Vertical Total	W
07h	Overflow	W
08h	Preset Row/Scan	W
09h	Maximum Scan Line	W
0Ah	Cursor Start	W
0Bh	Cursor End	W
0Ch	Start Address High	R/W
0Dh	Start Address Low	R/W
0Eh	Cursor Location High	R/W
0Fh	Cursor Location Low	R/W
10h	Lightpen High	R
10h	Vertical Sync Start	W
11h	Lightpen Low	R

Note: Indices 00h..18h apply to both 3B5h and 3D5h.

(Continued)

Table 3-5. CRT Controller Registers (Concluded)

Address	Register	Read/Write
(Continued):		
11h	Vertical Sync End	W
12h	Vertical Display End	W
13h	Offset	W
14h	Underline Location	W
15h	Start Vertical Blank	W
16h	End Vertical Blank	W
17h	Mode	W
18h	Line Compare	W

Note: Indices 00h..18h apply to both 3B5h and 3D5h.

Table 3-6. Attribute Controller Registers

Address	Register	Read/Write
3C0h	Index	W
3C0h	Data	W
00h	Palette 0	W
01h	Palette 1	W
02h	Palette 2	W
03h	Palette 3	W
04h	Palette 4	W
05h	Palette 5	W
06h	Palette 6	W
07h	Palette 7	W
08h	Palette 8	W
09h	Palette 9	W
0Ah	Palette 10	W
0Bh	Palette 11	W
0Ch	Palette 12	W
0Dh	Palette 13	W
0Eh	Palette 14	W
0Fh	Palette 15	W
10h	Mode Control	W
11h	Overscan Color	W
12h	Color Plane Enable	W
13h	Horizontal Pixel Panning	W

Table 3-7 lists the BIOS modes and their characteristics.

Table 3-7. BIOS Mode Operating Characteristics

BIOS Mode	Format/ Resolution	Operating Mode	Config.	No. of Colors Capable
0	40 x 25	Color	Text	16
1	40 x 25	Color	Text	16
2	80 x 25	Color	Text	16
3	80 x 25	Color	Text	16
4	320 x 200	Color	Compatible Graphics	4
5	320 x 200	Color	Compatible Graphics	4
6	640 x 200	Color	Compatible Graphics	2
7	80 x 25	Monochrome	Text	3 (Intensities)
8(1)	40 x 25	Color	Text	16
9(1)	40 x 25	Color	Text	16
A(1)	40 x 25	Color	Text	16
B(2)	80 x 25	Color	Text	N/A
C(2)	80 x 25	Monochrome	Text	N/A
D	320 x 200	Enhanced	Enhanced Graphics	16
E	640 x 200	Enhanced	Enhanced Graphics	16
F	640 x 350	Monochrome	Enhanced Graphics	3 (Intensities)
10	640 x 350	Enhanced	Enhanced Graphics	4 of 64
F+	640 x 350	Monochrome	Enhanced Graphics	3 (Intensities)
10+	640 x 350	Enhanced	Enhanced Graphics	16 of 64
0*	40 x 25	Color	Text	16 of 64
1*	40 x 25	Color	Text	16 of 64
2*	80 x 25	Color	Text	16 of 64
3*	80 x 25	Color	Text	16 of 64

Notes:
1. Identical to Mode 0; reserved for IBM® PC Jr.™ functionality.
2. Used to load character fonts; no video present when in this mode.
3. Throughout this chapter, references to BIOS Modes F and 10 are actually references to the full memory implementation of the monochrome graphics Mode F+ and color graphics Mode 10+. Because 256 KB of video RAM is standard on the COMPAQ Enhanced Color Graphics Board, Modes F+ and 10+ are always activated when Modes F and 10 are set, respectively.

Table 3-8 lists all the I/O addresses for the board.

Table 3-8. Listing of Specific I/O Addresses

Address	Register	Read/Write	Mode
3B4h	CRT Controller Index	W	Mono
3B5h	CRT Controller Data	R/W	Mono
3BAh	Input Status 1	R	Mono
3BAh	Feature Control	W	Mono
3C0h	Attribute Controller Index/ Data	W	C/M
3C2h	Input Status 0	R	C/M
3C2h	Miscellaneous Output	W	C/M
3C4h	Sequencer Index	W	C/M
3C5h	Sequencer Data	W	C/M
3C6h	Control Mode	R/W	C/M
3CAh	Graphics Controller 1 Position	W	C/M
3CCh	Graphics Controller 2 Position	W	C/M
3CEh	Graphics Controller Index	W	C/M
3CFh	Graphics Controller Data	W	C/M
3D4h	CRT Controller Index	W	Color
3D5h	CRT Controller Data	R/W	Color
3DAh	Input Status 1	R	Color
3DAh	Feature Control	W	Color
3DBh	Lightpen Latch Reset	W	Color
3DCh	Lightpen Latch Set	W	Color
7C6h	Environment	R	C/M
BC6h	Display Type	R	C/M
FC6h	Initial Mode	R	C/M

Legend: C/M = Color and monochrome

Table 3-9 lists all the video BIOS calls for the board.

Table 3-9. Video BIOS Calls

INT 05h	Print Screen
INT 0Ah	Vertical Blanking
INT 10h	Video I/O
AH=00h	Set Mode
AH=01h	Set Cursor Type
AH=02h	Set Cursor Position
AH=03h	Read Cursor Position
AH=04h	Read Lightpen Position
AH=05h	Select Active Display Page
AH=06h	Scroll Active Page Up
AH=07h	Scroll Active Page Down
AH=08h	Read Attribute/Character at Current Cursor Position
AH=09h	Write Attribute/Character at Current Cursor Position
AH=0Ah	Write Character Only at Current Cursor Position
AH=0Bh	Set Color Palette
AH=0Ch	Write Pixel
AH=0Dh	Read Pixel
AH=0Eh	Write TTY Character
AH=0Fh	Read Current Video State

(Continued)

Table 3-9. Video BIOS Calls (Continued)

AH=10h	Set Palette Register
AL=00h	Set Individual Palette Register
AL=01h	Set Overscan Color
AL=02h	Set All Palette Registers
AL=03h	Program Blink/Intensity
AH=11h	Load Character Generator
AL=00h	Load User Text Set
AL=01h	Load ROM Monochrome Set
AL=02h	Load ROM 8 x 8 Double-Dot Set
AL=03h	Set Block Specifier
AL=10h	Load User Text Set
AL=11h	Load ROM Monochrome Set
AL=12h	Load ROM 8 x 8 Double-Dot Set
AL=20h	Load User 8 x 8 Text Set
AL=21h	Load User Graphics Characters
AL=22h	Load ROM 8 x 14 Dot Set
AL=23h	Load ROM 8 x 8 Double-Dot Set
AL=30h	Return Font Information
AH=12h	Alternate Select
BL=10h	Return EGA Information
BL=20h	Select Alternate Print Screen Routine
AH=13h	Write TTY String
AH=BFh	Video BIOS Extensions
AL=00h	Switch to External Monitor
AL=01h	Switch to Internal Monitor
AL=02h	Change Control Mode
AL=03h	Read Video Environment
AL=04h	Set Mode-Switch Delay

(Continued)

Table 3-9. Video BIOS Calls (Concluded)

INT 1FH	Graphics Dot Table Vector
INT 42H	Compatible Video I/O
INT 43H	Graphics Dot Table Vector

TABLE OF CONTENTS

CHAPTER 4
REGISTER PROGRAMMING

4.1 I/O ADDRESSES 4-1

4.2 CONTROL AND STATUS REGISTERS 4-2

Input Status 0, Port 3C2h, Read Only 4-2
Miscellaneous Output, Port 3C2h, Write Only 4-3
Control Mode, Port 3C6h, Read/Write 4-3
Lightpen Latch Reset, Port 3DBh, Write Only 4-4
Lightpen Latch Set, Port 3DCh, Write Only 4-4
Input Status 1, Port 3xAh, Read Only 4-5
Feature Control, Port 3xAh, Write Only 4-7
Environment, Port 7C6h, Read Only 4-7
Display Type, Port BC6h, Read Only 4-8
Initial Mode, Port FC6h, Read Only 4-8

4.3 SEQUENCER REGISTERS 4-9

Sequencer Index, Port 3C4h, Write Only 4-9
Sequencer Reset, Port 3C5h, Index 00h, Write Only 4-10
Sequencer Clocking Mode, Port 3C5h, Index 01h, Write Only 4-10
Sequencer Write Plane Mask, Port 3C5h, Index 02h, Write Only 4-11
Sequencer Character Font Select, Port 3C5h, Index 03h, Write Only 4-12
Sequencer Memory Mode, Port 3C5h, Index 04h, Write Only 4-13

4.4 GRAPHICS CONTROLLER REGISTERS 4-13

Graphics Controller 2 Position, Port 3CAh, Write Only 4-14
Graphics Controller 1 Position, Port 3CCh, Write Only 4-14
Graphics Controller Index, Port 3CEh, Write Only 4-15
Graphics Controller Data Set/Reset, Port 3CFh, Index 00h, Write Only 4-15
Graphics Controller Enable Bit Set/Reset, Port 3CFh, Index 01h, Write Only 4-16
Graphics Controller Color Compare, Port 3CFh, Index 02h, Write Only 4-16
Graphics Controller Data Rotate, Port 3CFh, Index 03h, Write Only 4-17

Graphics Controller Read Plane Select, Port 3CFh, Index 04h, Write Only 4-17
Graphics Controller Mode, Port 3CFh, Index 05h, Write Only 4-18
Graphics Controller Miscellaneous, Port 3CFh, Index 06h, Write Only 4-19
Graphics Controller Color Don't Care, Port 3CFh, Index 07h, Write Only 4-20
Graphics Controller Bit Mask, Port 3CFh, Index 08h, Write Only 4-20
4.5 CRT CONTROLLER REGISTERS 4-21
CRT Controller Index, Port 3x4h, Write Only 4-23
CRT Controller Horizontal Total, Port 3x5h, Index 00h, Write Only 4-24
CRT Controller Horizontal Display End, Port 3x5h, Index 01h, Write Only 4-24
CRT Controller Start Horizontal Blank, Port 3x5h, Index 02h, Write Only 4-25
CRT Controller End Horizontal Blank, Port 3x5h, Index 03h, Write Only 4-25
CRT Controller Start Horizontal Sync, Port 3x5h, Index 04h, Write Only 4-26
CRT Controller End Horizontal Sync, Port 3x5h, Index 05h, Write Only 4-26
CRT Controller Vertical Total, Port 3x5h, Index 06h, Write Only 4-27
CRT Controller Overflow, Port 3x5h, Index 07h, Write Only 4-27
CRT Controller Preset Row Scan, Port 3x5h, Index 08h, Write Only 4-28
CRT Controller Maximum Scan Line, Port 3x5h, Index 09h, Write Only 4-28
CRT Controller Cursor Start, Port 3x5h, Index 0Ah, Write Only 4-29
CRT Controller Cursor End, Port 3x5h, Index 0Bh, Write Only 4-29
CRT Controller Start Address High, Port 3x5h, Index 0Ch, Read/Write 4-30
CRT Controller Start Address Low, Port 3x5h, Index 0Dh, Read/Write 4-31
CRT Controller Cursor Location High, Port 3x5h, Index 0Eh, Read/Write 4-31
CRT Controller Cursor Location Low, Port 3x5h, Index 0Fh, Read/Write 4-32
CRT Controller Lightpen High, Port 3x5h, Index 10h, Read Only 4-32
CRT Controller Vertical Sync Start, Port 3x5h, Index 10h, Write Only 4-33
CRT Controller Lightpen Low, Port 3x5h, Index 11h, Read Only 4-33
CRT Controller Vertical Sync End, Port 3x5h, Index 11h, Write Only 4-34
CRT Controller Vertical Display End, Port 3x5h, Index 12h, Write Only 4-34

TABLE OF CONTENTS (Continued)

CRT Controller Offset, Port 3x5h, Index 13h, Write Only 4-35
CRT Controller Underline Location, Port 3x5h, Index 14h, Write Only 4-35
CRT Controller Start Vertical Blank, Port 3x5h, Index 15h, Write Only 4-36
CRT Controller End Vertical Blank, Port 3x5h, Index 16h, Write Only 4-36
CRT Controller Mode Control, Port 3x5h, Index 17h, Write Only 4-37
CRT Controller Line Compare, Port 3x5h, Index 18h, Write Only 4-38
4.6 ATTRIBUTE CONTROLLER REGISTERS 4-38
Attribute Controller Index, Port 3C0h, Write Only 4-39
Attribute Controller Palette 0..15, Port 3C0h, Index 00h..0Fh, Write Only 4-40
Attribute Controller Mode Control, Port 3C0h, Index 10h, Write Only 4-40
Attribute Controller Overscan Color, Port 3C0h, Index 11h, Write Only 4-41
Attribute Controller Color Plane Enable, Port 3C0h, Index 12h, Write Only 4-42
Attribute Controller Horizontal Pixel Panning, Port 3C0h, Index 13h, Write Only 4-43
4.7 MODE-SPECIFIC REGISTER VALUES 4-43

Chapter 4
REGISTER PROGRAMMING

This chapter describes in detail the various registers available on the COMPAQ Enhanced Color Graphics Board. It provides programming information, such as addresses for the registers and bit maps that illustrate bit values and functions. The sections are:

- I/O Addresses
- Control and Status Registers
- Sequencer Registers
- Graphics Controller Registers
- CRT Controller Registers
- Attribute Controller Registers
- Mode-Specific Register Values

4.1 I/O ADDRESSES

Table 4-1 lists I/O addresses for the bit registers of the COMPAQ Enhanced Color Graphics Board. Many of the addresses contain a lowercase "x", which indicates that the function associated with that address is mode dependent. In color modes, "x" equals "D". In monochrome modes, "x" equals "B". This rule also applies to the other tables in this section.

Table 4-1. I/O Addresses for Registers

I/O Address	R/W	Register Name
3C0h	W	Attribute Controller Index
3C0h	W	Attribute Controller Data
3C2h	R W	Input Status 0 Miscellaneous Output
3C4h	W	Sequencer Index
3C5h	W	Sequencer Data
3C6h (See Note)	R/W	Control Mode (COMPAQ Specific)
3CAh	W	Graphics Controller 2 Position
3CCh	W	Graphics Controller 1 Position
3CEh	W	Graphics Controller Index
3CFh	W	Graphics Controller Data
3x4h	W	CRT Controller Index
3x5h	R/W	CRT Controller Data
3xAh	R W	Input Status 1 Feature Control
3DBh	W	Lightpen Latch Reset
3DCh	W	Lightpen Latch Set
7C6h (Note)	R	Environment
BC6h (Note)	R	Display Type
FC6h (Note)	R	Initial Mode

Note: These COMPAQ-specific registers decode all 16 bits in I/O space; therefore, bits <15..12> must equal 0.

4.2 CONTROL AND STATUS REGISTERS

Table 4-2 lists the Control and Status registers and their addresses.

Table 4-2. Control and Status Registers

Address	Register	Read/Write
3C2h	Input Status 0	R
3C2h	Miscellaneous Output	W
3C6h (Note)	Control Mode	R/W
3xAh	Input Status 1	R
3xAh	Feature Control	W
3xBh	Lightpen Latch Reset	W
3xCh	Lightpen Latch Set	W
7C6h (Note)	Environment	R
BC6h (Note)	Display Type	R
FC6h (Note)	Initial Mode	R

Note: These COMPAQ-specific registers decode all 16 bits in I/O space; therefore, bits <15..12> must equal 0.

INPUT STATUS 0

PORT 3C2h, READ ONLY

This register contains the status of the SW1 switch, the Feature Status signals, and the Vertical Retrace Interrupt status. Parenthesized pin numbers refer to Special Feature connector J4.

BIT
76543210

0000 (Not Used)

Multiplexed switch SW1 sense (see Misc. Output register, bits < 3..2>)
0 = Switch closed (on)
1 = Switch open (off)

Feature Status 0
0 = Feat 0 (Pin 19) low
1 = Feat 0 (Pin 19) high or open

Feature Status 1
0 = Feat 1 (Pin 17) low
1 = Feat 1 (Pin 17) high or open

Vertical Retrace Interrupt (IRQ2) status (See Note)
0 = Interrupt armed
1 = Interrupt initiated

Note: Bit <7> is valid only when vertical retrace interrupts are enabled. In all other cases, it is set to 1.

MISCELLANEOUS OUTPUT
PORT 3C2h, WRITE ONLY

This register controls miscellaneous functions of the COMPAQ Enhanced Color Graphics Board.

```
    BIT
76543210
||||||||
|||||||└─ I/O Base Address
|||||||    0 = 3Bxh, monochrome display
|||||||    1 = 3Dxh, color display
||||||└── CPU access to video memory
||||||     1 = Disabled
||||||     0 = Enabled
||||└┴─── Master Clock Frequency and SW1 Select
||||       00 =  Frequency 1 (See Table 4-3),
||||             SW1, position 4
||||       01 =  Frequency 2 (See Table 4-3),
||||             SW1, position 3
||||       10 =  External, SW1, position 2
||||       11 =  Not used, SW1, position 1
|||└────── 0 = Video drivers enabled
|||        1 = Special Features interface enabled
||└─────── Page Select Bit in Odd/Even Mode
||         0 = Low page
||         1 = High page
|└──────── Horizontal Sync Polarity
|          0 = (+)
|          1 = (-)
└───────── Vertical Sync Polarity
           0 = (+)
           1 = (-)
```

CONTROL MODE
PORT 3C6h, READ/WRITE

This register controls the operating mode and selection of monitors.

```
    BIT
76543210
||||||└┴─ Video Controller Mode
||||||     00 = Reserved
||||||     01 = EGA Mode
||||||     10 = Reserved
||||||     11 = Reserved
|||||└─── Reserved (Always 0)
||||└──── Internal/External Select
||||       0 = External monitor active
||||       1 = Internal monitor active
|||└───── External Display Blank (See Note)
|||        0 = Enables normal hardware
|||            blanking logic
|||        1 = Blanks external monitor
|||            signal
||└────── Internal Display Blank (See Note)
||         0 = Enables normal hardware
||             blanking logic
||         1 = Blanks internal monitor
||             signal
|└─────── Restore Mode
|          0 = Disables
|          1 = Enables
└──────── Diagnostic Mode
           0 = Enables normal hardware blanking
           1 = Disables hardware blanking
```

Note: Bits <5..4> should not be used for monitor blanking (for example, SCRNSAVE), because they are used by the hardware compatibility logic.

LIGHTPEN LATCH RESET
PORT 3DBh, WRITE ONLY

Writing any value to port 3DBh resets the lightpen latch. The state of the latch is reflected in bit <1> of the Input Status 1 register.

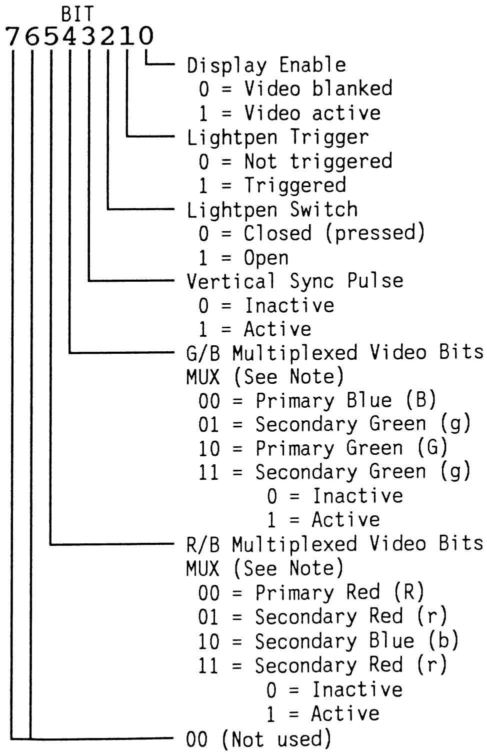
LIGHTPEN LATCH SET
PORT 3DCh, WRITE ONLY

Writing any value to port 3DCh resets the lightpen latch. The state of the latch is reflected in bit <1> of the Input Status 1 register.

INPUT STATUS 1
PORT 3xAh, READ ONLY

This register contains the status of various video signals.

Use the DISPLAY ENABLE signal bit as an indicator of active display times. This signal combines the horizontal and vertical sync and other blanking periods, and indicates when the display is active.



Display Enable
0 = Video blanked
1 = Video active

Lightpen Trigger
0 = Not triggered
1 = Triggered

Lightpen Switch
0 = Closed (pressed)
1 = Open

Vertical Sync Pulse
0 = Inactive
1 = Active

G/B Multiplexed Video Bits
MUX (See Note)
00 = Primary Blue (B)
01 = Secondary Green (g)
10 = Primary Green (G)
11 = Secondary Green (g)
0 = Inactive
1 = Active

R/B Multiplexed Video Bits
MUX (See Note)
00 = Primary Red (R)
01 = Secondary Red (r)
10 = Secondary Blue (b)
11 = Secondary Red (r)
0 = Inactive
1 = Active

00 (Not used)

Note: MUX = Bits <5..4> of the Color Plane Enable register in the attribute controller.

Table 4-3 shows how master clock frequencies (in megahertz) are a function of mode and monitors. Frequencies are set in bits <3..2> of the Miscellaneous Output register. Formats of 640 x 200 include 320 x 200 and formats of 640 x 350 include 320 x 350.

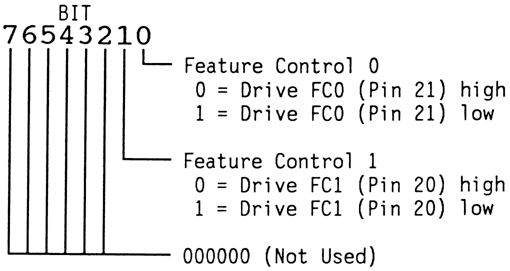
Table 4-3. Master Clock Frequencies

Mode	Monitor	Format 1	Frequency 1 (MHz)	Format 2	Frequency 2 (MHz)
Color	COMPAQ Color	640 x 200	14.318	640 x 350	16.257
	COMPAQ Dual-Mode	640 x 200	14.318	640 x 350	16.872
	RGBI Color	640 x 200	14.318	-	-
Mono	COMPAQ Dual-Mode	640 x 350	16.872	720 x 350	18.981

FEATURE CONTROL

PORT 3xAh, WRITE ONLY

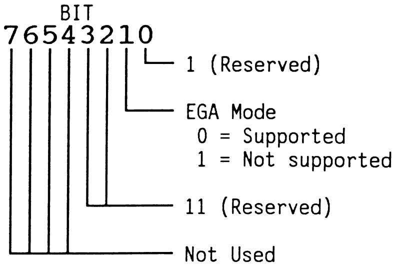
This register specifies the state of the FC0 and FC1 signals. Parenthesized pin numbers refer to Special Feature connector J4.


BIT
76543210
Feature Control 0
0 = Drive FC0 (Pin 21) high
1 = Drive FC0 (Pin 21) low
Feature Control 1
0 = Drive FC1 (Pin 20) high
1 = Drive FC1 (Pin 20) low
000000 (Not Used)

ENVIRONMENT

PORT 7C6h, READ ONLY

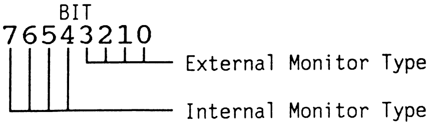
This register can be read to determine whether a COMPAQ Video Display Controller Board supports the EGA mode.


BIT
76543210
1 (Reserved)
EGA Mode
0 = Supported
1 = Not supported
11 (Reserved)
Not Used

DISPLAY TYPE
PORT BC6h, READ ONLY

This register indicates whether an external and/or internal monitor is installed, and what type it is.

The setting of configuration switch bank SW2 is reflected in the Display Type register, port BC6h. On SW2, the sense of positions 4, 3, 2, and 1 are read in bits <5..4> and <1..0>, respectively. Bits <7..6> and <3..2> always return a 0. A switch in the OFF position returns a 1. The standard setting is SW2, position 4 = ON.


BIT
76543210
External Monitor Type
Internal Monitor Type

Monitor Type

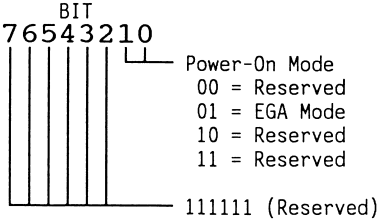
0000 = None
0001 = COMPAQ Dual-Mode (DM) Monitor
0010 = RGBI Color Monitor (See Note)
0011 = COMPAQ Color Monitor (RrGgBb) or Compatible
0100 = Reserved
. .
. .
1111 = Reserved

Note: An internal RGBI color monitor is not supported.

INITIAL MODE
PORT FC6h, READ ONLY

This register indicates that the board is going to operate in EGA mode when the system is reinitialized from either a power ON/OFF cycle or warm boot (CTRL + ALT + DEL).

The settings of switch SW2, positions 6 and 5, are reflected in bits <1..0>, respectively. A switch in the OFF position returns a 1. The standard setting is SW2, position 6 = ON, position 5 = OFF.


BIT
76543210
Power-On Mode
00 = Reserved
01 = EGA Mode
10 = Reserved
11 = Reserved
111111 (Reserved)

4.3 SEQUENCER REGISTERS

Table 4-4 lists the Sequencer registers, their addresses, and their indices.

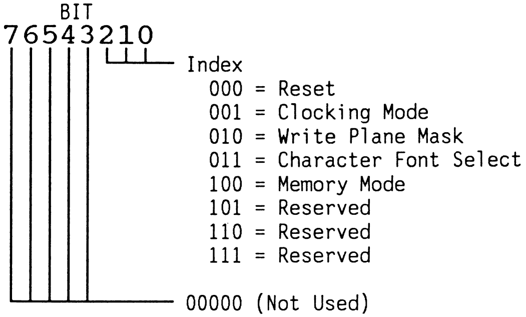
Table 4-4. Sequencer Registers

Address	Register	Read/Write
3C4h	Index	W
3C5h	Data	W
00h	Reset	W
01h	Clocking Mode	W
02h	Write Plane Mask	W
03h	Character Font Select	W
04h	Memory Mode	W

The sequencer uses two I/O (port) addresses for register access. Port 3C4h is the Sequencer Index register. Port 3C5h interacts with one of five Sequencer registers pointed to by the Sequencer Index register.

SEQUENCER INDEX
PORT 3C4h, WRITE ONLY

This register points to one of five Sequencer registers.

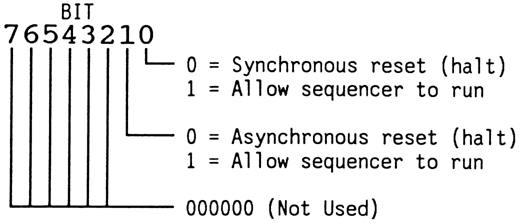


000 = Reset
001 = Clocking Mode
010 = Write Plane Mask
011 = Character Font Select
100 = Memory Mode
101 = Reserved
110 = Reserved
111 = Reserved

00000 (Not Used)

SEQUENCER RESET
PORT 3C5h, INDEX 00h, WRITE ONLY

This register resets the sequencer.



0 = Synchronous reset (halt)
1 = Allow sequencer to run

0 = Asynchronous reset (halt)
1 = Allow sequencer to run

000000 (Not Used)

In a synchronous reset, the sequencer waits until a "synchronized" point when it can halt without any possibility of data loss.

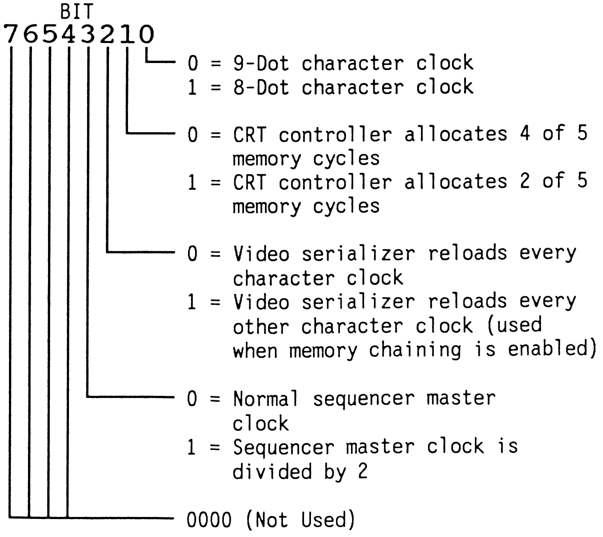
In an asynchronous reset, the sequencer immediately "locks" all the registers and counters. Stopping with this bit can cause data loss in the video RAM.

Bits <0> and <1> must both be set to 1 for the sequencer to run.

SEQUENCER CLOCKING MODE
PORT 3C5h, INDEX 01h, WRITE ONLY

This register specifies clock rates for various sequencer functions.

To change the contents of the Sequencer Clocking Mode register, synchronously halt the sequencer, make the change, then restart the sequencer.



0 = 9-Dot character clock
1 = 8-Dot character clock

0 = CRT controller allocates 4 of 5 memory cycles
1 = CRT controller allocates 2 of 5 memory cycles

0 = Video serializer reloads every character clock
1 = Video serializer reloads every other character clock (used when memory chaining is enabled)

0 = Normal sequencer master clock
1 = Sequencer master clock is divided by 2

0000 (Not Used)

The monochrome text mode is the only mode that uses a 9-dot-wide character clock.

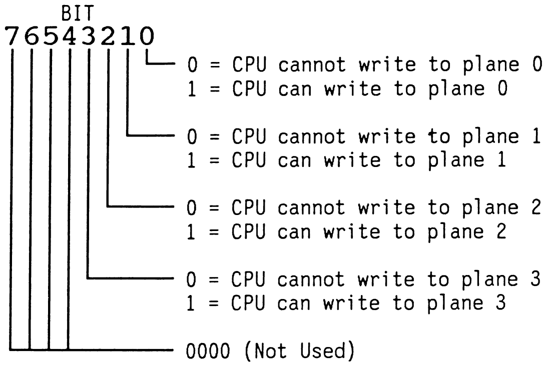
All high-resolution modes (80-column text or 640-pixel graphics or greater) require that the CRT controller get four out of every five memory cycles for display refresh.

The dot clock bit (bit <3>) generates the 320-dot modes by halving the master clock input.

SEQUENCER WRITE PLANE MASK

PORT 3C5h, INDEX 02h, WRITE ONLY

This register specifies which planes of video memory are disabled.



0 = CPU cannot write to plane 0
1 = CPU can write to plane 0

0 = CPU cannot write to plane 1
1 = CPU can write to plane 1

0 = CPU cannot write to plane 2
1 = CPU can write to plane 2

0 = CPU cannot write to plane 3
1 = CPU can write to plane 3

0000 (Not Used)

When odd/even modes are selected, planes 0 and 1 and 2 and 3 should have the same Plane Mask values.

SEQUENCER CHARACTER FONT SELECT
PORT 3C5h, INDEX 03h, WRITE ONLY

This register specifies which two fonts are used at any one time as the source of the dot patterns for the character generator. In text mode, plane 2 is divided into four 16-KB banks. Each bank contains one 256-character font.

This register also specifies which 16-KB bank of video memory is the source of the dot patterns for the character generator.

BIT
76543210

Bits <1..0> specify which 16-KB bank to use when the text-attribute byte, bit <3> = 0.
00 = First 16 KB
01 = Second 16 KB
10 = Third 16 KB
11 = Fourth 16 KB

Bits <3..2> specify which 16-KB bank to use when the text-attribute byte, bit <3> = 1.
00 = First 16 KB
01 = Second 16 KB
10 = Third 16 KB
11 = Fourth 16 KB

0000 (Not Used)

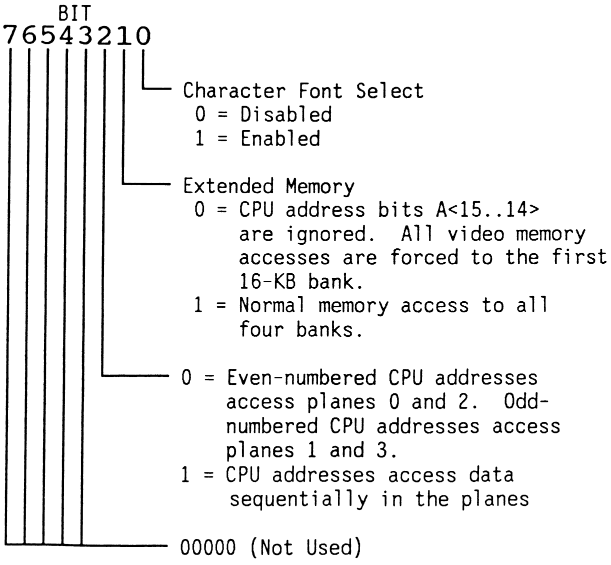
Bit <3> of the text-attribute byte specifies whether the text (foreground) is in highlight or whether an alternate character font (shown above) is used. To allow bit <3> of the text-attribute byte to select dual-character sets, set bit <1> and the text bit <0> in the Sequencer Memory Mode register; otherwise font 0 is always selected.

To maintain consistent colors when using alternate fonts, some palette registers for the intensified colors 8..15 must be reloaded for non-intensified colors, or the contents of the Color Plane Enable register must be changed to ignore plane 3.

SEQUENCER MEMORY MODE

PORT 3C5h, INDEX 04h, WRITE ONLY

This register controls CPU access to the video memory and enables the Character Font Select function, allowing dual-character sets.



Character Font Select
0 = Disabled
1 = Enabled

Extended Memory
0 = CPU address bits A<15..14> are ignored. All video memory accesses are forced to the first 16-KB bank.
1 = Normal memory access to all four banks.

0 = Even-numbered CPU addresses access planes 0 and 2. Odd-numbered CPU addresses access planes 1 and 3.
1 = CPU addresses access data sequentially in the planes

00000 (Not Used)

Bit <2> (odd/even bit) controls ONLY the CPU write accesses. CPU read accesses are controlled by the odd/even bit in the Graphics Controller Mode register.

4.4 GRAPHICS CONTROLLER REGISTERS

Table 4-5 lists the Graphics Controller registers, their addresses, and their indices.

Table 4-5. Graphics Controller Registers

Address	Register	Read/Write
3CAh	Graphics Controller 2 Position	W
3CCh	Graphics Controller 1 Position	W
3CEh	Index	W
3CFh	Data	W
00h	Data Set/Reset	W
01h	Enable Bit Set/Reset	W
02h	Color Compare	W
03h	Data Rotate	W
04h	Read Plane Select	W
05h	Mode	W
06h	Miscellaneous	W
07h	Color Don't Care	W
08h	Bit Mask	W

GRAPHICS CONTROLLER 2 POSITION
PORT 3CAh, WRITE ONLY

This register specifies which bits of the CPU data bus the Graphics Controller 2 controls.

```
    BIT
76543210
||||| └┴── 00 = CPU data bits 0 and 1
|||||      01 = CPU data bits 2 and 3
|||||      10 = CPU data bits 4 and 5
|||||      11 = CPU data bits 6 and 7
|||||
└┴┴┴┴───── 000000 (Not Used)
```

Port 3CAh, bits <1..0> should always be set to 01.

GRAPHICS CONTROLLER 1 POSITION
PORT 3CCh, WRITE ONLY

This register specifies which bits of the CPU data bus the Graphics Controller 1 controls.

```
    BIT
76543210
||||| └┴── 00 = CPU data bits 0 and 1
|||||      01 = CPU data bits 2 and 3
|||||      10 = CPU data bits 4 and 5
|||||      11 = CPU data bits 6 and 7
|||||
└┴┴┴┴───── 000000 (Not Used)
```

Port 3CCh, bits <1..0> should always be set to 00.

GRAPHICS CONTROLLER INDEX
PORT 3CEh, WRITE ONLY

Port 3CEh serves as the Index register for both graphics controllers.

BIT
76543210

Bits 3-0: Index
- 0000 = Data Set/Reset
- 0001 = Enable Bit Set/Reset
- 0010 = Color Compare
- 0011 = Data Rotate
- 0100 = Read Plane Select
- 0101 = Mode
- 0110 = Miscellaneous
- 0111 = Color Don't Care
- 1000 = Bit Mask
- 1001 = Reserved
- . .
- . .
- 1111 = Reserved

Bits 7-4: 0000 (Not Used)

GRAPHICS CONTROLLER DATA SET/RESET
PORT 3CFh, INDEX 00h, WRITE ONLY

This register stores a bit pattern to write to video memory when in Write Mode 0 (see the Graphics Controller Mode register). Values are written to each plane unless access to that plane is disabled.

BIT
76543210

Bit 0: Data bit in plane 0
- 0 = Reset
- 1 = Set

Bit 1: Data bit in plane 1
- 0 = Reset
- 1 = Set

Bit 2: Data bit in plane 2
- 0 = Reset
- 1 = Set

Bit 3: Data bit in plane 3
- 0 = Reset
- 1 = Set

Bits 7-4: 0000 (Not Used)

The graphics controller must be in Write Mode 0 for the Data Set/Reset function to perform. The logic operation selected by the Graphics Controller Data Rotate register is also applied to the output of the Data Set/Reset register. Output from this register is not written to a plane unless that plane is enabled in the Sequencer Write Plane Mask register and the Graphics Controller Enable Bit Set/Reset register.

GRAPHICS CONTROLLER ENABLE BIT SET/RESET
PORT 3CFh, INDEX 01h, WRITE ONLY

This register enables or disables the Bit Set/Reset function on a plane-by-plane basis. When the Bit Set/Reset function is disabled, CPU data is written to video memory. When the Bit Set/Reset function is enabled, the value in the Data Set/Reset register is written to video memory.

```
   BIT
76543210
||||||└── Bit Set/Reset for plane 0
||||||      0 = Disabled
||||||      1 = Enabled
|||||└─── Bit Set/Reset for plane 1
|||||       0 = Disabled
|||||       1 = Enabled
||||└──── Bit Set/Reset for plane 2
||||        0 = Disabled
||||        1 = Enabled
|||└───── Bit Set/Reset for plane 3
|||         0 = Disabled
|||         1 = Enabled
└┴┴────── 0000 (Not Used)
```

The graphics controller must be in Write Mode 0 for the Bit Set/Reset function to perform. Any logic operations selected are also performed on the Data Set/Reset register output.

GRAPHICS CONTROLLER COLOR COMPARE
PORT 3CFh, INDEX 02h, WRITE ONLY

This register contains a 4-bit plane pattern (color definition) that is used during a color comparison (Read Mode 1). In Read Mode 1, normal reads cause a color comparison between the contents of this register and the video memory addressed by the read. Each pixel (dot) color in the byte read is compared to the color value and a 1 is returned in the corresponding bit position if the colors match. Otherwise, a 0 is returned when a match does not occur in the corresponding bit position.

```
   BIT
76543210
|||└┴┴┴── These 4 bits define the color to
|||       compare with the colors in video
|||       memory.
└┴┴────── 0000 (Not Used)
```

Read Mode 1 must be in effect for the Graphics Controller Color Compare register to be used. The Color Compare feature in Read Mode 1 is not affected by the Graphics Controller Read Plane Select register; however, it is affected by the Graphics Controller Color Don't Care register.

GRAPHICS CONTROLLER DATA ROTATE
PORT 3CFh, INDEX 03h, WRITE ONLY

This register specifies the number of bits to rotate data and/or the type of logic operation to perform on data during a CPU write to video memory.

BIT
76543210

These bits specify the number of bits to rotate data. This is a bit rotation to the right.

These bits determine the type of logic operation to apply between CPU or Data Set/Reset register output before the data is written to video memory.

- 00 = Source data unmodified
- 01 = Source data ANDed with latch data
- 10 = Source data ORed with latch data
- 11 = Source data XORed with latch data

000 (Not Used)

Data rotation is performed before the logic operation and is done only in Write Mode 0. When the data source is the latched data (Write Mode 1), the logic operation is not applied and latch data is unmodified.

GRAPHICS CONTROLLER READ PLANE SELECT
PORT 3CFh, INDEX 04h, WRITE ONLY

This register determines the bit plane read. This determination occurs by loading a value into bits <2..0>, depending on the input conditions (CPU address line A0 and Graphics Controller Mode register odd/even bit <4>). This register and these input conditions are valid in Read Mode 0 only.

BIT
76543210

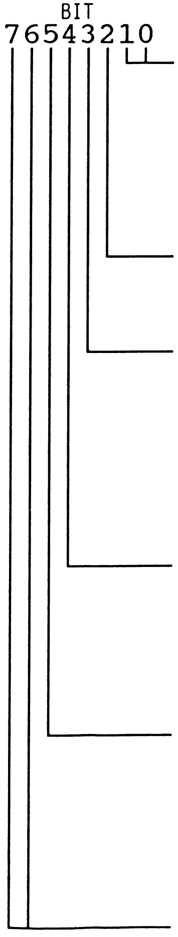
Plane Select	Input Conditions CPU Add. Line A0	Odd/ Even	Result (Plane Read)
000	X	0	0
001	X	0	1
010	X	0	2
011	X	0	3
1XX	X	0	None
000	0	1	0
001	0	1	0
010	0	1	2
011	0	1	2
1XX	0	1	None
000	1	1	1
001	1	1	1
010	1	1	3
011	1	1	3
1XX	1	1	None

00000 (Reserved)

Legend: X = Don't care

GRAPHICS CONTROLLER MODE
PORT 3CFh, INDEX 05h, WRITE ONLY

This register defines the current operating modes for the graphics controller.



00 = Write Mode 0. CPU or Data Set/Reset register output is written to video memory
01 = Write Mode 1. Latch data is written to video memory
10 = Write Mode 2. Plane n is filled with data bit <n>.
11 = Reserved

0 = Normal graphics controller signal output
1 = Graphics controller output signals are tri-stated

0 = Read Mode 0. The CPU reads the memory plane selected by the Graphics Controller Read Plane Select register.
1 = Read Mode 1. The CPU reads the results of the comparison between the video memory byte and the Graphics Controller Color Compare register.

Odd/even bit
0 = CPU reads data sequentially from the planes
1 = Even CPU addresses access planes 0 and 2
Odd CPU addresses access planes 1 and 3

0 = Video output data is serialized and sent to the attribute controller
1 = Video output data is serialized with even-numbered bits from even-numbered planes and odd-numbered bits from odd-numbered planes

00 (Not Used)

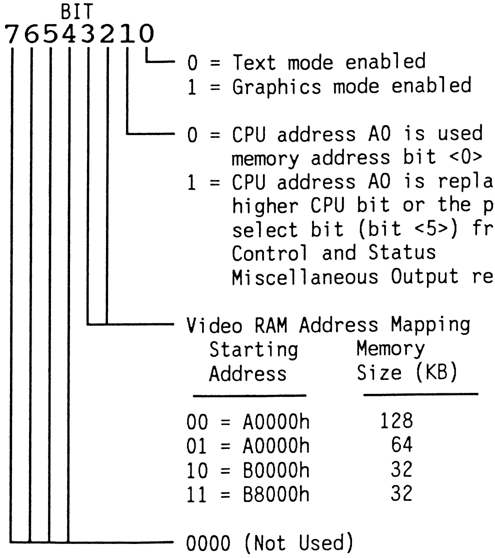
The rotate count from the Graphics Controller Data Rotate register is applied to the CPU data in Write Mode 0.
The logic function specified in the Data Rotate register is applied to the data and latches in all write modes except Write Mode 1. Data is not affected in Write Mode 1.

Bits not selected by the Graphics Controller Bit Mask register receive their value from the latches. The latches are loaded when the CPU performs a video memory read operation.

Bit <5> when set to 1 facilitates the bit packing required by BIOS Modes 4 and 5 (320 x 200).

GRAPHICS CONTROLLER MISCELLANEOUS
PORT 3CFh, INDEX 06h, WRITE ONLY

This register defines some miscellaneous functions of the graphics controller.



0 = Text mode enabled
1 = Graphics mode enabled

0 = CPU address A0 is used as video memory address bit <0>
1 = CPU address A0 is replaced by a higher CPU bit or the page select bit (bit <5>) from the Control and Status Miscellaneous Output register

Video RAM Address Mapping

	Starting Address	Memory Size (KB)
00 =	A0000h	128
01 =	A0000h	64
10 =	B0000h	32
11 =	B8000h	32

0000 (Not Used)

If the video display controller is mapped at A0000h and the video memory size is 128 KB, no other video display controller board can be in the system (video memory conflict occurs).

GRAPHICS CONTROLLER COLOR DON'T CARE
PORT 3CFh, INDEX 07h, WRITE ONLY

This register specifies which planes are ignored during a color comparison between the Graphics Controller Color Compare register and the value in video memory. When a plane is ignored, the comparison result is always a match for that plane.

This register is used only when the graphics controller is in Read Mode 1.

BIT
76543210

Bit 0:
0 = Plane 0 is ignored
1 = Plane 0 is included in the comparison

Bit 1:
0 = Plane 1 is ignored
1 = Plane 1 is included in the comparison

Bit 2:
0 = Plane 2 is ignored
1 = Plane 2 is included in the comparison

Bit 3:
0 = Plane 3 is ignored
1 = Plane 3 is included in the comparison

Bits 7-4:
0000 (Not Used)

GRAPHICS CONTROLLER BIT MASK
PORT 3CFh, INDEX 08h, WRITE ONLY

This register contains a mask pattern that determines whether CPU data or graphics controller latch data is written to video memory.

Bit rotations or logic operations, if any, occur before the masking operation and before the write to video memory takes place.

Bits set to 0 in this register cause the corresponding graphics controller latch bit to be written to video memory. Bits set to 1 in this register cause the corresponding CPU data bit to be written to video memory.

4.5 CRT CONTROLLER REGISTERS

The CRT controller is a powerful, programmable, large-scale-integrated circuit. It generates the majority of the signals required to produce a video display. The programmability of the CRT controller allows it to operate with a variety of monitors and with different operating modes.

The CRT controller contains internal working registers and counters that are not accessible by the CPU. The values in the CRT controller's counters are continually compared with the values written to the indexed registers. When the counter value coincides with, or matches, the value of the indexed register, a signal or process begins or ends.

The majority of CRT registers control the placement of the display on the screen (centering, number of characters, scan lines, amount of blanking, and so on).

Some register descriptions give the register contents in terms of a variable value n. For some registers, you are instructed to subtract a constant value from the value for n and place the resulting value in the register. This process is required for proper operation of the CRT controller.

Note: If the CRT controller is programmed in a 350-scan line mode and a COMPAQ Dual-Mode Monitor is being driven, some values will be substituted by the hardware compatibility logic. Therefore, nonstandard BIOS values may not react as expected.

Table 4-6 lists the CRT controller registers, their addresses, and their indices.

Table 4-6. CRT Controller Registers

Address	Register	Read/Write
3x4h	Index	-
3x5h	Data	-
00h	Horizontal Total	W
01h	Horizontal Display End	W
02h	Start Horizontal Blank	W
03h	End Horizontal Blank	W
04h	Start Horizontal Sync	W
05h	End Horizontal Sync	W
06h	Vertical Total	W
07h	Overflow	W
08h	Preset Row/Scan	W
09h	Maximum Scan Line	W
0Ah	Cursor Start	W
0Bh	Cursor End	W
0Ch	Start Address High	R/W
0Dh	Start Address Low	R/W
0Eh	Cursor Location High	R/W
0Fh	Cursor Location Low	R/W
10h	Lightpen High	R
10h	Vertical Sync Start	W
11h	Lightpen Low	R
11h	Vertical Sync End	W
12h	Vertical Display End	W
13h	Offset	W
14h	Underline Location	W
15h	Start Vertical Blank	W
16h	End Vertical Blank	W
17h	Mode	W
18h	Line Compare	W

CRT CONTROLLER INDEX
PORT 3x4h, WRITE ONLY

This register serves as a pointer to the various CRT controller internal registers. The Index register can be addressed as port 3B4h for the monochrome mode, and 3D4h for the color/graphics modes.

```
    BIT
76543210
| | | └┴┴┴┴── Value indicates destination register
| | |         of next CRT controller access
| | |
└─┴─┴──────── 000 (Don't Care)
```

Table 4-7 lists the values for bits <4..0> and their associated functions.

Table 4-7. CRT Controller Indices

	Bits						
Index	4	3	2	1	0	R/W	Register Function
00h	0	0	0	0	0	W	Horizontal total
01h	0	0	0	0	1	W	Horizontal display end
02h	0	0	0	1	0	W	Start horizontal blank
03h	0	0	0	1	1	W	End horizontal blank
04h	0	0	1	0	0	W	Start horizontal sync
05h	0	0	1	0	1	W	End horizontal sync
06h	0	0	1	1	0	W	Vertical total
07h	0	0	1	1	1	W	Overflow
08h	0	1	0	0	0	W	Preset row scan
09h	0	1	0	0	1	W	Maximum scan line
0Ah	0	1	0	1	0	W	Cursor start
0Bh	0	1	0	1	1	W	Cursor end
0Ch	0	1	1	0	0	R/W	Start address high (MSB)
0Dh	0	1	1	0	1	R/W	Start address low (LSB)
0Eh	0	1	1	1	0	R/W	Cursor location high (MSB)
0Fh	0	1	1	1	1	R/W	Cursor location low (LSB)
10h	1	0	0	0	0	W	Vertical sync start
10h	1	0	0	0	0	R	Lightpen high (MSB)
11h	1	0	0	0	1	W	Vertical sync end
11h	1	0	0	0	1	R	Lightpen low (LSB)
12h	1	0	0	1	0	W	Vertical display end
13h	1	0	0	1	1	W	Offset
14h	1	0	1	0	0	W	Underline location
15h	1	0	1	0	1	W	Start vertical blank
16h	1	0	1	1	0	W	End vertical blank
17h	1	0	1	1	1	W	Mode control
18h	1	1	0	0	0	W	Line compare

CRT CONTROLLER HORIZONTAL TOTAL
PORT 3x5h, INDEX 00h, WRITE ONLY

This register defines the total number of character times in a horizontal scan interval, including the retrace period. If n is the total number of character times, then the contents of this register should be n - 2.

```
   BIT
76543210
└┴┴┴┴┴┴┴─── Total number of character times (n)
            minus 2
```

CRT CONTROLLER HORIZONTAL DISPLAY END
PORT 3x5h, INDEX 01h, WRITE ONLY

This register determines the duration of the HORIZONTAL DISPLAY ENABLE signal as the total number of displayed characters. If n is the total number of character times, then the contents of this register should be n - 1.

```
   BIT
76543210
└┴┴┴┴┴┴┴─── Total number of character times,
            (n) minus 1
```

CRT CONTROLLER START HORIZONTAL BLANK
PORT 3x5h, INDEX 02h, WRITE ONLY

This register specifies the point where horizontal blanking begins, in character times.

```
   BIT
76543210
└┴┴┴┴┴┴┴──── Horizontal blanking begins when
             these bits coincide with the value in
             the internal horizontal-character
             counter
```

CRT CONTROLLER END HORIZONTAL BLANK
PORT 3x5h, INDEX 03h, WRITE ONLY

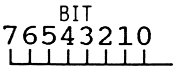
This register specifies when horizontal blanking ends, in character times, and specifies a skew amount for the DE (display enable) signal.

```
    BIT
76543210
│││└┴┴┴┴──── Horizontal blanking ends when these
│││          5 bits coincide with the last
│││          5 bits of the horizontal-
│││          character counter
││
│└┴───────── Skew of the  DE (display enable)
│            signal, in character times
│             00 = 0
│             01 = 1
│             10 = 2
│             11 = 3
│
└─────────── 0 (Not Used)
```

The maximum blanking signal width (difference between blank start and end) is 31 character times.

CRT CONTROLLER START HORIZONTAL SYNC
PORT 3x5h, INDEX 04h, WRITE ONLY

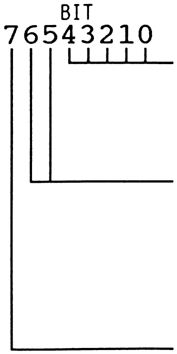
This register specifies the starting point for the horizontal sync period, in character times. This register centers the display horizontally by changing the horizontal sync position.



The horizontal sync period begins when these bits coincide with the value in the internal horizontal-character counter

CRT CONTROLLER END HORIZONTAL SYNC
PORT 3x5h, INDEX 05h, WRITE ONLY

This register specifies the point where the horizontal sync period ends, in character times, and specifies a skew amount for the horizontal sync signal.



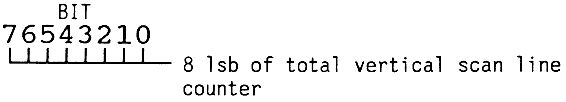
The horizontal sync period ends when these 5 bits coincide with the last 5 bits of the horizontal-character counter

Skew (right direction) of the horizontal sync signal
00 = 0
01 = 1
10 = 2
11 = 3

0 = Video data loaded at end of horizontal sync period
1 = Video data loaded one character time before end of horizontal sync period. This bit is used when the video serializers are set to load every other character clock. (Modes F and 10 [NOT F+ or 10+].) Sync width is limited to 31 characters.

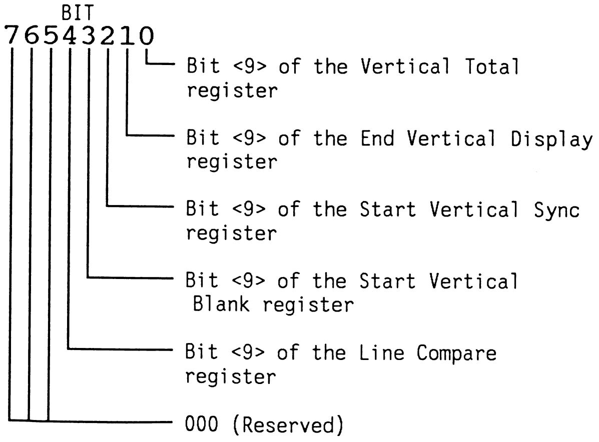
CRT CONTROLLER VERTICAL TOTAL
PORT 3x5h, INDEX 06h, WRITE ONLY

This register contains the 8 least-significant bits (lsb) of the 9-bit total number of horizontal scans in a vertical interval (display plus retrace). Bit <9> is defined in the CRT Controller Overflow register.


BIT
76543210
8 lsb of total vertical scan line counter

CRT CONTROLLER OVERFLOW
PORT 3x5h, INDEX 07H, WRITE ONLY

This register contains bit <9>, the most-significant bit (msb) of other CRT controller registers.


BIT
76543210
Bit <9> of the Vertical Total register
Bit <9> of the End Vertical Display register
Bit <9> of the Start Vertical Sync register
Bit <9> of the Start Vertical Blank register
Bit <9> of the Line Compare register
000 (Reserved)

CRT CONTROLLER PRESET ROW SCAN
PORT 3x5h, INDEX 08h, WRITE ONLY

This register specifies the first displayed scan line after a vertical retrace.

```
     BIT
76543210
| |  └┴┴┴┴── The display is enabled when these 5
| |           bits coincide with the last 5 bits
| |           of the scan line counter
| |
└─┴────────── 000 (Not Used)
```

This register can be used to implement smooth pixel scrolling in the vertical direction. The maximum preset row scan is limited to 31 pixels (5 bits). This value should NOT exceed the maximum scan line value or unpredictable results may occur. The value in the Preset Row Scan register is latched in the CRT controller at the start of vertical retrace. Therefore, this register should be updated by software prior to the start of the vertical retrace period.

CRT CONTROLLER MAXIMUM SCAN LINE
PORT 3x5h, INDEX 09h, WRITE ONLY

This register defines the number of scan lines per character. If _n_ equals the number of scan lines per character, then load this register with _n_ - 1.

```
     BIT
76543210
| |  └┴┴┴┴── Number of scan lines per character
| |           n minus 1
| |
└─┴────────── 000 (Not Used)
```

Character height is limited to 32 scan lines.

CRT CONTROLLER CURSOR START
PORT 3x5h, INDEX 0Ah, WRITE ONLY

This register defines the starting scan line for the cursor. If the starting scan line exceeds the maximum scan line, then the cursor is invisible.

```
    BIT
76543210
|||└┴┴┴┴── The cursor scan lines begin when
|||        these bits coincide with the last
|||        5 bits of the character cell scan
|||        line counter
|||
└┴┴─────── 000 (Not Used)
```

CRT CONTROLLER CURSOR END
PORT 3x5h, INDEX 0Bh, WRITE ONLY

This register specifies the last scan line for the cursor, and specifies a skew amount for the cursor signal. If *n* equals the ending scan line value for the cursor, then load bits <4..0> of the Cursor End register with *n* + 1. When *n* + 1 exceeds the maximum scan line (as defined in the Maximum Scan Line register) and the starting scan line value is nonzero, then load bits <0..4> with 0. See Table 4-8.

Table 4-8. Typical Cursor Register Programming

Cursor Scan Lines for 8 x 14 Character Cell		Data Loaded Into Bits <4..0> of Either the Cursor Start or the Cursor End Register	
Starting Scan Line	Ending Scan Line *n*	Cursor Start (See Note)	Cursor End
0	13	00000	01110
11	12	01011	01101
12	13	01100	00000

Note: The data for the Cursor Start register is given for reference purposes.

BIT
76543210

The cursor ends when these 5 bits coincide with the 5 lsb of the character scan line counter

Skew of the cursor signal, in character times
00 = 0
01 = 1
10 = 2
11 = 3

0 (Reserved)

CRT CONTROLLER START ADDRESS HIGH
PORT 3x5h, INDEX 0Ch, READ/WRITE

This register specifies the 8 most-significant bits of the 16-bit starting address of displayed video memory. The 8 least-significant bits are stored in the Start Address Low register, Index 0Dh. (See Note.)

BIT
76543210

Most-significant byte of video memory start address

Note: The values in the Start Address High and Start Address Low registers are latched in the CRT controller at the start of vertical retrace. Therefore, these registers should be updated by software prior to the start of the vertical retrace period.

CRT CONTROLLER START ADDRESS LOW
PORT 3x5h, INDEX 0Dh, READ/WRITE

This register specifies the 8 least-significant bits of the 16-bit starting address of displayed video memory. The 8 most-significant bits are stored in the Start Address High register, Index 0Ch. (See Note.)

```
   BIT
76543210
|_|_|_|_|_|_|_|_|__ Least-significant byte of video
                    memory start address
```

Note: The values in the Start Address High and Start Address Low registers are latched in the CRT controller at the start of vertical retrace. Therefore, these registers should be updated by software prior to the start of the vertical retrace period.

CRT CONTROLLER CURSOR LOCATION HIGH
PORT 3x5h, INDEX 0Eh, READ/WRITE

This register defines the 8 most-significant bits of the 16-bit video memory address for the cursor. The 8 least-significant bits of this register are in the Cursor Location Low register, Index 0Fh.

```
   BIT
76543210
|_|_|_|_|_|_|_|_|__ Most-significant byte of the cursor
                    video memory address
```

CRT CONTROLLER CURSOR LOCATION LOW
PORT 3x5h, INDEX 0Fh, READ/WRITE

This register defines the 8 least-significant bits of the 16-bit video memory address for the cursor. The 8 most-significant bits of this register are in the Cursor Location High register, Index 0Eh.

```
   BIT
76543210
└┴┴┴┴┴┴┴── Least-significant byte of the cursor
           video memory address
```

CRT CONTROLLER LIGHTPEN HIGH
PORT 3x5h, INDEX 10h, READ ONLY

This register returns the 8 most-significant bits of the video memory address, which is latched when the lightpen strobe signal becomes active. The 8 least-significant bits are returned in the Lightpen Low register, Index 11h.

```
   BIT
76543210
└┴┴┴┴┴┴┴── Most-significant byte of the video
           memory address
```

CRT CONTROLLER VERTICAL SYNC START

PORT 3x5h, INDEX 10h, WRITE ONLY

This register contains the 8 least-significant bits of the 9-bit value that specifies the starting scan line for the vertical sync period.

This value can be used to center the screen vertically by changing the vertical sync position. Bit <9> of this value is located in the Overflow register.

```
   BIT
76543210
└┴┴┴┴┴┴┴┴── The vertical sync  period begins
            when the 9-bit Vertical Sync
            Start address bits coincide with
            the last 9 bits of the scan line
            counter
```

CRT CONTROLLER LIGHTPEN LOW

PORT 3x5h, INDEX 11h, READ ONLY

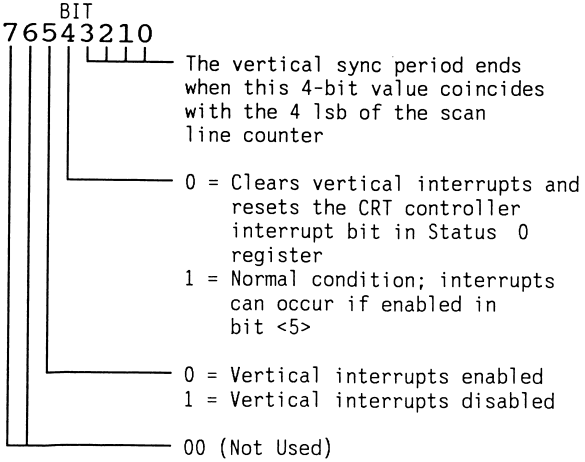
This register returns the 8 least-significant bits of the video memory address that is latched when the lightpen strobe signal becomes active. The 8 most-significant bits are returned in the Lightpen High register, Index 10h.

```
   BIT
76543210
└┴┴┴┴┴┴┴┴── Least-significant byte of the video
            memory address
```

CRT CONTROLLER VERTICAL SYNC END
PORT 3x5h, INDEX 11h, WRITE ONLY

This register contains the 4-bit value that specifies the ending scan line for the vertical sync period.

Vertical interrupts occur at the start of the vertical-blanking period, not the start of the vertical sync period. Interrupts from the CRT controller occur on the IRQ2 line or IRQ9 (on 80286- or 80386-based products).



Bits 3–0: The vertical sync period ends when this 4-bit value coincides with the 4 lsb of the scan line counter

Bit 4:
0 = Clears vertical interrupts and resets the CRT controller interrupt bit in Status 0 register
1 = Normal condition; interrupts can occur if enabled in bit <5>

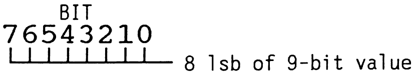
Bit 5:
0 = Vertical interrupts enabled
1 = Vertical interrupts disabled

Bits 7–6: 00 (Not Used)

CRT CONTROLLER VERTICAL DISPLAY END
PORT 3x5h, INDEX 12h, WRITE ONLY

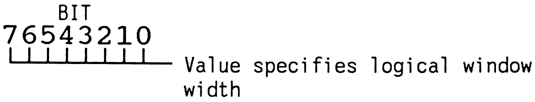
This register contains the 8 least-significant bits of the 9-bit value that specifies the total number of displayed scan lines. If n equals the total number of displayed scan lines, then load this register with n - 1.

Bit <9> is stored in the Overflow register.



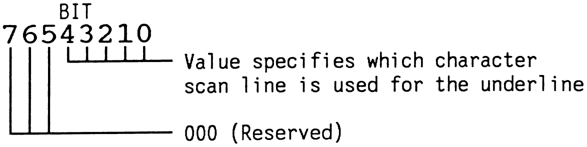
CRT CONTROLLER OFFSET
PORT 3x5h, INDEX 13h, WRITE ONLY

This register specifies a skew value for the display window. These bits define the logical line width or logical window size. The starting memory address for the next displayable row, character, or scan line is greater than the current row address by this amount. The value is in words or double-words (dword) based on the CRT clocking mode. This register is used in conjunction with the Preset Row Scan and the Horizontal Panning registers to provide smooth horizontal and vertical panning.


BIT
76543210
Value specifies logical window width

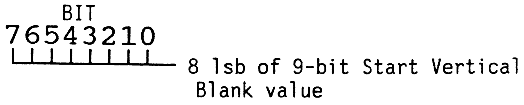
CRT CONTROLLER UNDERLINE LOCATION
PORT 3x5h, INDEX 14h, WRITE ONLY

This register specifies which character scan line is used for the underline. Underlining is active in all text modes, including the color graphics text mode. When the character scan line specified for underlining is set to greater than the maximum scan line, no underline occurs.


BIT
76543210
Value specifies which character scan line is used for the underline
000 (Reserved)

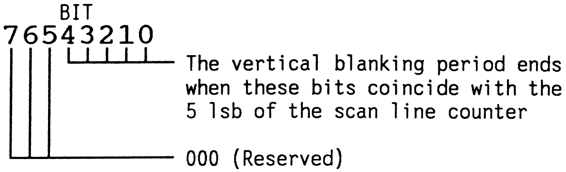
CRT CONTROLLER START VERTICAL BLANK
PORT 3x5h, INDEX 15h, WRITE ONLY

This register contains the 8 least-significant bits of the 9-bit Start Vertical Blank value. Bit <9> is stored in the Overflow register. Vertical blanking begins when the 9-bit value coincides with the last 9 bits of the scan line counter.


BIT
76543210
8 lsb of 9-bit Start Vertical Blank value

CRT CONTROLLER END VERTICAL BLANK
PORT 3x5h, INDEX 16h, WRITE ONLY

This register specifies the ending scan line of the vertical blanking period. The maximum blanking-signal width is 31 scan lines because of the 5-bit limitation.


BIT
76543210
The vertical blanking period ends when these bits coincide with the 5 lsb of the scan line counter
000 (Reserved)

CRT CONTROLLER MODE CONTROL
PORT 3x5h, INDEX 17h, WRITE ONLY

This register defines various CRT controller operating modes.

BIT
76543210

0 = Substitute address bit <0> for bit <13> during active display cycles
1 = No substitution

0 = Substitute address bit <1> for bit <14> during active display cycles
1 = No substitution

0 = Scan line counter increments once per horizontal sync pulse (normal)
1 = Scan line counter increments after every other horizontal sync pulse

0 = Display memory address increments once per character clock pulse
1 = Display memory address increments after every other character clock pulse

0 = CRT controller outputs are enabled
1 = CRT controller outputs are tri-stated

Address wrap controls address bit <0> in word mode
0 = Selects address bit <13> as lsb
1 = Selects address bit <15> as lsb

Word/byte mode select controls the memory addressing mode
0 = Selects word mode, shifts all memory addresses up by 1 and replaces bit <0> with msb
1 = Selects byte addressing mode and addressing is passed through

0 = Horizontal and vertical syncs are disabled
1 = Horizontal and vertical syncs are enabled

CRT CONTROLLER LINE COMPARE
PORT 3x5h, INDEX 18h, WRITE ONLY

This register contains the 8 least-significant bits of a 9-bit value that specifies a scan line count where the video memory address is to be reset to zero. This register generates horizontally split screens, or windows, on the display.

For normal non-split display, this 9-bit value should be set to its maximum (1FFh). Bit <9> of this register is located in the Overflow register.

```
  BIT
76543210
└┴┴┴┴┴┴┴─ 8 lsb of the 9-bit scan Line
          Compare value
```

4.6 ATTRIBUTE CONTROLLER REGISTERS

The Attribute Controller Index and Data registers are accessed through a single port. An internal flip-flop in the attribute controller determines whether the Index register or the Data register is being accessed. Accesses to this port (3C0h) alternate between the Index and Data registers. Before accessing port 3C0h, reset the internal flip-flop to a known state by reading port 3xAh (Input Status 1 register). The first write to the attribute controller is to the Index register.

Table 4-9 lists the attribute controller registers, their addresses, and their indices.

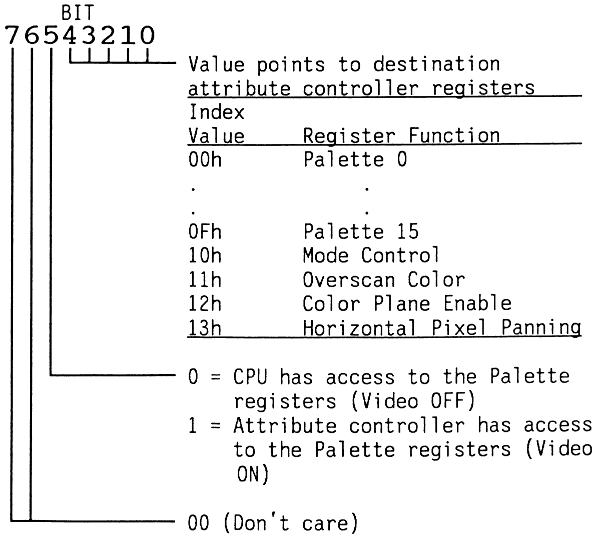
Table 4-9. Attribute Controller Registers

Address	Register	Read/Write
3C0h	Index	-
3C0h	Data	-
00h	Palette 0	W
01h	Palette 1	W
02h	Palette 2	W
03h	Palette 3	W
04h	Palette 4	W
05h	Palette 5	W
06h	Palette 6	W
07h	Palette 7	W
08h	Palette 8	W
09h	Palette 9	W
0Ah	Palette 10	W
0Bh	Palette 11	W
0Ch	Palette 12	W
0Dh	Palette 13	W
0Eh	Palette 14	W
0Fh	Palette 15	W
10h	Mode Control	W
11h	Overscan Color	W
12h	Color Plane Enable	W
13h	Horizontal Pixel Panning	W

ATTRIBUTE CONTROLLER INDEX
PORT 3C0h, WRITE ONLY

This is an Index register that points to other attribute controller registers.

If the Palette registers are accessed by the system, all the color outputs go to zero, thereby blanking the display.



Value points to destination attribute controller registers

Index Value	Register Function
00h	Palette 0
.	.
.	.
0Fh	Palette 15
10h	Mode Control
11h	Overscan Color
12h	Color Plane Enable
13h	Horizontal Pixel Panning

0 = CPU has access to the Palette registers (Video OFF)
1 = Attribute controller has access to the Palette registers (Video ON)

00 (Don't care)

Video can be turned off by manipulating bit <5>.

ATTRIBUTE CONTROLLER PALETTE 0..15
PORT 3C0h, INDEX 00h..0Fh, WRITE ONLY

Sixteen, 6-bit registers comprise the color palette. Each register specifies what color (or monochrome intensity) is to be displayed for a given attribute or color code. Depending on the operating mode, the Palette register bits can specify one of 64 colors on an enhanced color (RrGgBb) monitor, one of 16 colors on an RGBI color monitor, or, when operating in a monochrome mode, one of three intensities on a COMPAQ Dual-Mode Monitor.

The Palette registers should be modified only during blanking periods to avoid "glitches" in the display.

BIT 76543210	Operating Mode		
	Enhanced Color	RGBI Color	Monochrome
0	Primary Blue	Blue	0
1	Primary Green	Green	0
2	Primary Red	Red	0
3	Secondary Blue	0	Video
4	Secondary Green	Intense	Intense
5	Secondary Red	0	0
7, 6	00 (Reserved)	00 (Reserved)	00 (Reserved

ATTRIBUTE CONTROLLER MODE CONTROL
PORT 3C0h, INDEX 10h, WRITE ONLY

This register defines various attribute controller operating modes.

In the text modes, the intensity/blank select bit changes the meaning of bit <7> of the character-attribute byte, so that bit <7> is either the background intensity bit or the blink bit.

In the graphics modes, the intensity/blink select bit changes the meaning of bit <3>, so that it is either part of the color information or the blink bit.

The blink rate is fixed at 32 vertical periods.

BIT
76543210

- Bit 0:
  - 0 = Attribute controller treats data as text
  - 1 = Attribute controller treats data as graphics
- Bit 1:
  - 0 = Attribute byte interpreted as color attributes
  - 1 = Attribute byte interpreted as monochrome attributes
- Bit 2: In the 9-dot mode (monochrome),
  - 0 = 9th dot matches background
  - 1 = 9th dot is duplicate of 8th dot for ASCII codes C0h..DFh
- Bit 3: Intensity/Blink Bit:

  In the text modes,
  - 0 = Intensity attribute selected
  - 1 = Blink attribute selected

  In the graphics modes,
  - 0 = Color attribute selected
  - 1 = Blink attribute selected
- Bits 7-4: 0000 (Reserved)

ATTRIBUTE CONTROLLER OVERSCAN COLOR
PORT 3C0h, INDEX 11h, WRITE ONLY

This register defines a border or overscan color.

The 350-scan line modes have almost no overscan area and, therefore, almost no border area. The Overscan Color register should be modified only during the vertical retrace period.

BIT
76543210

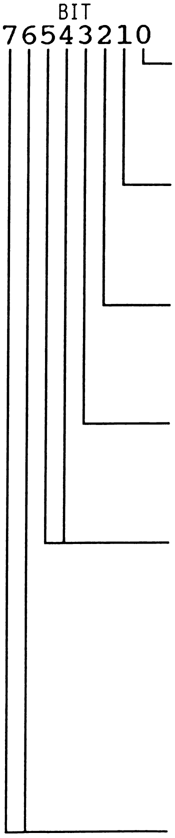
- Bit 0: Blue (B)
  - 0 = Active
  - 1 = Inactive
- Bit 1: Green (G)
  - 0 = Active
  - 1 = Inactive
- Bit 2: Red (R)
  - 0 = Active
  - 1 = Inactive
- Bit 3: Secondary Blue (b)
  - 0 = Active
  - 1 = Inactive
- Bit 4: Secondary Green (g)/Intensity (I)
  - 0 = Active
  - 1 = Inactive
- Bit 5: Secondary Red (r)
  - 0 = Active
  - 1 = Inactive
- Bits 7-6: 00 (Reserved)

ATTRIBUTE CONTROLLER COLOR PLANE ENABLE
PORT 3C0h, INDEX 12h, WRITE ONLY

This register controls access to the color planes and selects which color planes are read from Input Status 1 register.

If bit <4> of this register is set, all six color outputs will be tri-stated (all-white video); however, the values gated to Input Status 1 register will be correct, because they are gated before the tri-state buffers.

The values loaded into Input Status 1 register are gated after the blanking logic; during a blanking period the gated values are blanked (reset to zeros) also.



Attribute controller access to plane 0
0 = Disabled
1 = Enabled

Attribute controller access to plane 1
0 = Disabled
1 = Enabled

Attribute controller access to plane 2
0 = Disabled
1 = Enabled

Attribute controller access to plane 3
0 = Disabled
1 = Enabled

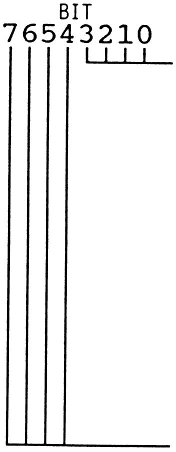
Video bit MUX select. These two bits control which 2 video bits are reflected in bits <5> and <4> of Input Status 1 register.

	Input Status 1 Register	
MUX	5	4
00 =	Pri. Red	Pri. Blue
01 =	Sec. Red	Sec. Green
10 =	Sec. Blue	Pri. Green
11 =	Sec. Red	Sec. Green

00 (Not Used)

ATTRIBUTE CONTROLLER HORIZONTAL PIXEL PANNING
PORT 3C0h, INDEX 13h, WRITE ONLY

This register specifies the number of pixels to shift the display to the left in any display mode. This is not a wraparound shift.



Values range from 0 to 7 for 8-dot display modes, and from 0 to 8 for 9-dot modes

8-Dot Mode		9-Dot Mode	
Value	Pixels Shifted	Value	Pixels Shifted
0	0	0	8
1	1	1	0
2	2	2	1
3	3	3	2
4	4	4	3
5	5	5	4
6	6	6	5
7	7	7	6
		8	7

0000 (Reserved)

The Horizontal Pixel Panning register should be updated only during the vertical retrace period.

4.7 MODE-SPECIFIC REGISTER VALUES

This section contains tables showing the specific values written by the BIOS to the registers for each of the BIOS modes. These values are provided for reference purposes for the systems developer who needs working examples of register programming. Before changing any of the default parameters in the registers, be sure that the environment is understood (that is, know the types of monitors connected to the controller and the timing that each display requires) for proper operation.

If the applications software is operating in a single-tasking operating system environment, it is strongly recommended that operating modes be set by making calls to BIOS INT 10h using AH=00h (Set Mode). Adopting this practice lessens the dependence of the software on the particular type of video controller installed and monitor(s) attached.

Table 4-10 shows the initial register values for the BIOS modes. Lightpen register values, cursor positions, and other read-only register values are not shown.

Table 4-10. Initial Register Values (In Hexadecimal Notation)

				BIOS Mode																						
Register Group/Name	Port	Index	R/W	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	10	F+	10+	0*	1*	2*	3*
Control and Status Registers:																										
Miscellaneous Output	3C2h	--	W	23	23	23	23	23	23	23	A6	23	23	23	23	A6	23	23	A2	A7	A2	A7	A7	A7	A7	A7
Control Mode	3C6h	--	W	NP	NP	NP	NP	NP	NP	NP	NP	NP	NP	NP	NP	NP	NP	NP	NP	NP	NP	NP	NP	NP	NP	NP
Feature Control	3xAh	--	W	NP	NP	NP	NP	NP	NP	NP	NP	NP	NP	NP	NP	NP	NP	NP	NP	NP	NP	NP	NP	NP	NP	NP
Sequencer Registers:																										
Index	3C4h	--	W	IX	IX	IX	IX	IX	IX	IX	IX	IX	IX	IX	IX	IX	IX	IX	IX	IX	IX	IX	IX	IX	IX	IX
Reset	3C5h	00h	W	03	03	03	03	03	03	03	03	03	03	03	03	03	03	03	03	03	03	03	03	03	03	03
Clocking Mode	3C5h	01h	W	0B	0B	01	01	0B	0B	01	00	00	00	00	01	00	0B	01	05	05	01	01	0B	0B	01	01
Write Plane Mask	3C5h	02h	W	03	03	03	03	03	03	01	03	00	00	00	04	04	0F	0F	0F	0F	0F	0F	03	03	03	03
Character Font Select	3C5h	03h	W	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
Memory Mode	3C5h	04h	W	03	03	03	03	02	02	06	03	03	03	03	07	07	06	06	00	00	06	06	03	03	03	03

Legend

3xAh - This is the Feature Control/Input Status 1 register address, which is dependent on the mode of the adapter. The following shows the possible address values and the corresponding modes:
3BAh - If the adapter is in the monochrome mode.
3DAh - If the adapter is in the color mode.

NP - This register is not programmed during the setting of a BIOS mode.

IX - This is an index register that determines which of the other registers is accessed. This register changes continually during the programming process to access the other registers.

(Continued)

Table 4-10. Initial Register Values (In Hexadecimal Notation) (Continued)

				BIOS Mode																						
Register Group/Name	Port	Index	R/W	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	10	F+	10+	0*	1*	2*	3*
Graphics Controller Registers:																										
Graphics 2 Position	3CAh	--	W	01	01	01	01	01	01	01	01	01	01	01	01	01	01	01	01	01	01	01	01	01	01	01
Graphics 1 Position	3CCh	--	W	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
Graphics 1 & 2 Index	3CEh	--	W	IX	IX	IX	IX	IX	IX	IX	IX	IX	IX	IX	IX	IX	IX	IX	IX	IX	IX	IX	IX	IX	IX	IX
Data Set/Reset	3CFh	00h	W	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
Enable Bit Set/Reset	3CFh	01h	W	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
Color Compare	3CFh	02h	W	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
Data Rotate	3CFh	03h	W	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
Read Plane Select	3CFh	04h	W	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
Mode	3CFh	05h	W	10	10	10	10	30	30	00	10	10	10	10	00	00	00	00	10	10	00	00	10	10	10	10
Miscellaneous	3CFh	06h	W	0E	0E	0E	0E	0F	0F	0D	0A	0E	0E	0E	04	04	05	05	07	07	05	05	0E	0E	0E	0E
Color Don't Care	3CFh	07h	W	00	00	00	00	00	00	00	00	00	00	00	00	00	0F	0F	0F	0F	0F	0F	00	00	00	00
Bit Mask	3CFh	08h	W	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF

Legend

IX - This is an index register that determines which of the other registers is accessed. This register changes continually during the programming process to access the other registers.

(Continued)

Table 4-10. Initial Register Values (In Hexadecimal Notation) (Continued)

				BIOS Mode																						
Register Group/Name	Port	Index	R/W	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	10	F+	10+	0*	1*	2*	3*
CRT Controller Registers:																										
Index	3x4h	--	W	IX	IX	IX	IX	IX	IX	IX	IX	IX	IX	IX	IX	IX	IX	IX	IX	IX	IX	IX	IX	IX	IX	IX
Horizontal Total	3x5h	00h	W	37	37	70	70	37	37	70	60	37	37	37	70	60	37	70	60	5B	60	5B	2D	2D	5B	5B
Horizontal Display End	3x5h	01h	W	27	27	4F	4F	27	27	4F	4F	27	27	27	4F	4F	27	4F	4F	4F	4F	4F	27	27	4F	4F
Start Horiz. Blank	3x5h	02h	W	2D	2D	5C	5C	2D	2D	59	56	2D	2D	2D	5C	56	2D	59	56	53	56	53	2B	2B	53	53
End Horiz. Blank	3x5h	03h	W	37	37	2F	2F	37	37	2D	3A	37	37	37	2F	3A	37	2D	1A	17	3A	37	2D	2D	37	37
Start Horiz. Sync	3x5h	04h	W	31	31	5F	5F	30	30	5E	51	31	31	31	5F	51	30	5E	50	50	50	52	28	28	51	51
End Horiz. Sync	3x5h	05h	W	15	15	07	07	14	14	06	60	15	15	15	07	60	14	06	E0	BA	60	00	6D	6D	5B	5B
Vertical Total	3x5h	06h	W	04	04	04	04	04	04	04	70	04	04	04	04	70	04	04	70	6C	70	6C	6C	6C	6C	6C
Overflow	3x5h	07h	W	11	11	11	11	11	11	11	1F	11	11	11	11	1F	11	11	1F	1F	1F	1F	1F	1F	1F	1F
Preset Row/Scan	3x5h	08h	W	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
Maximum Scan Line	3x5h	09h	W	07	07	07	07	01	01	01	0D	07	07	07	07	0D	00	00	00	00	00	00	0D	0D	0D	0D
Cursor Start	3x5h	0Ah	W	06	06	06	06	00	00	00	0B	06	06	06	06	0B	00	00	00	00	00	00	06	06	06	06
Cursor End	3x5h	0Bh	W	07	07	07	07	00	00	00	0C	07	07	07	07	0C	00	00	00	00	00	00	07	07	07	07

Legend

3x4h - This is the CRT Controller Index register address, which is dependent on the mode of the adapter. The following shows the possible address values and corresponding mode:
3B4h - If the adapter is in the monochrome mode.
3D4h - If the adapter is in the color mode.

3x5h - This is the CRT Controller Data register address, which is dependent on the mode of the adapter. The following shows the possible address values and corresponding mode:
3B5h - If the adapter is in the monochrome mode.
3D5h - If the adapter is in the color mode.

IX - This is an index register that determines which of the other registers is accessed. This register changes continually during the programming process to access the other registers.

(Continued)

Table 4-10. Initial Register Values (In Hexadecimal Notation) (Continued)

Register Group/Name	Port	Index	R/W	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	10	F+	10+	0*	1*	2*	3*
CRT Controller Registers (Continued):																										
Start Address High	3x5h	0Ch	R/W	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
Start Address Low	3x5h	0Dh	R/W	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
Cursor Location High	3x5h	0Eh	R\W	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
Cursor Location Low	3x5h	0Fh	R\W	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
Vertical Sync Start	3x5h	10h	W	E1	E1	E1	E1	E1	E1	E0	5E	E1	E1	E1	E1	5E	E1	E0	5E	5E	5E	5E	5E	5E	5E	5E
Vertical Sync End	3x5h	11h	W	24	24	24	24	24	24	23	2E	24	24	24	24	2E	24	23	2E	2B	2E	2B	2B	2B	2B	2B
Vertical Display End	3x5h	12h	W	C7	C7	C7	C7	C7	C7	C7	5D	C7	C7	C7	C7	5D	C7	C7	5D	5D	5D	5D	5D	5D	5D	5D
Offset	3x5h	13h	W	14	14	28	28	14	14	28	28	14	14	14	28	28	14	28	14	14	28	28	14	14	28	28
Underline Location	3x5h	14h	W	08	08	08	08	00	00	00	0D	08	08	08	08	0D	00	00	0D	0F	0D	0F	0F	0F	0F	0F
Start Vertical Blank	3x5h	15h	W	E0	E0	E0	E0	E0	E0	DF	5E	E0	E0	E0	E0	5E	E0	DF	5E	5F	5E	5F	5E	5E	5E	5E
End Vertical Blank	3x5h	16h	W	F0	F0	F0	F0	F0	F0	EF	6E	F0	F0	F0	F0	6E	F0	EF	6E	0A	6E	0A	0A	0A	0A	0A
Mode	3x5h	17h	W	A3	A3	A3	A3	A2	A2	C2	A3	A3	A3	A3	A3	A3	E3	E3	8B	8B	E3	E3	A3	A3	A3	A3
Line Compare	3x5h	18h	W	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF

Legend

3x5h - This is the CRT Controller Data register address, which is dependent on the mode of the adapter. The following shows the possible address values and corresponding mode:

3B5h - If the adapter is in the monochrome mode.
3D5h - If the adapter is in the color mode.

(Continued)

Table 4-10. Initial Register Values (In Hexadecimal Notation) (Concluded)

				BIOS Mode																						
Register Group/Name	Port	Index	R/W	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	10	F+	10+	0*	1*	2*	3*
Attribute Controller Registers:																										
Index	3C0h	--	W	IX	IX	IX	IX	IX	IX	IX	IX	IX	IX	IX	IX	IX	IX	IX	IX	IX	IX	IX	IX	IX	IX	IX
Palette 0	3C0h	00h	W	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
Palette 1	3C0h	01h	W	01	01	01	01	13	13	17	08	01	01	01	00	00	01	01	08	00	08	01	01	01	01	01
Palette 2	3C0h	02h	W	02	02	02	02	15	15	17	08	02	02	02	00	00	02	02	00	00	00	02	02	02	02	02
Palette 3	3C0h	03h	W	03	03	03	03	17	17	17	08	03	03	03	00	00	03	03	00	00	00	03	03	03	03	03
Palette 4	3C0h	04h	W	04	04	04	04	02	02	17	08	04	04	04	00	00	04	04	18	04	18	04	04	04	04	04
Palette 5	3C0h	05h	W	05	05	05	05	04	04	17	08	05	05	05	00	00	05	05	18	07	18	05	05	05	05	05
Palette 6	3C0h	06h	W	06	06	06	06	06	06	17	08	06	06	06	00	00	06	06	00	00	00	14	14	14	14	14
Palette 7	3C0h	07h	W	07	07	07	07	07	07	17	08	07	07	07	00	00	07	07	00	00	00	07	07	07	07	07
Palette 8	3C0h	08h	W	10	10	10	10	10	10	17	10	10	10	10	00	00	10	10	00	00	00	38	38	38	38	38
Palette 9	3C0h	09h	W	11	11	11	11	11	11	17	18	11	11	11	00	00	11	11	08	01	08	39	39	39	39	39
Palette 10	3C0h	0Ah	W	12	12	12	12	12	12	17	18	12	12	12	00	00	12	12	00	00	00	3A	3A	3A	3A	3A
Palette 11	3C0h	0Bh	W	13	13	13	13	13	13	17	18	13	13	13	00	00	13	13	00	00	00	3B	3B	3B	3B	3B
Palette 12	3C0h	0Ch	W	14	14	14	14	14	14	17	18	14	14	14	00	00	14	14	00	04	00	3C	3C	3C	3C	3C
Palette 13	3C0h	0Dh	W	15	15	15	15	15	15	17	18	15	15	15	00	00	15	15	18	07	18	3D	3D	3D	3D	3D
Palette 14	3C0h	0Eh	W	16	16	16	16	16	16	17	18	16	16	16	00	00	16	16	00	00	00	3E	3E	3E	3E	3E
Palette 15	3C0h	0Fh	W	17	17	17	17	17	17	17	18	17	17	17	00	00	17	17	00	00	00	3F	3F	3F	3F	3F
Mode Control	3C0h	10h	W	08	08	08	08	01	01	01	0E	08	08	08	00	0E	01	01	0B	01	0B	01	08	08	08	08
Overscan Color	3C0h	11h	W	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
Color Plane Enable	3C0h	12h	W	0F	0F	0F	0F	03	03	01	0F	0F	0F	0F	0F	0F	0F	0F	05	05	05	0F	0F	0F	0F	0F
Horiz. Pixel Panning	3C0h	13h	W	00	00	00	00	00	00	00	08	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00

Legend

IX - This is an index register that determines which of the other registers is accessed. This register changes continually during the programming process to access the other registers.

TABLE OF CONTENTS

CHAPTER 5
BIOS

5.1 INTRODUCTION TO VIDEO BIOS 5-1
5.2 VIDEO BIOS MODES SUPPORTED 5-2
5.3 DEFAULT COLOR PALETTE 5-4
5.4 INITIALIZATION 5-6
5.5 POWER-ON SELF-TEST 5-6
5.6 INTERRUPTS 5-6
INT 05h - SW - Print Screen 5-7
INT 0Ah - HW - IRQ2, Vertical Blanking 5-8
INT 10h - SW - Video I/O 5-9
INT 10h, AH = 00h - Set Mode 5-11
INT 10h, AH = 01h - Set Cursor Type 5-12
INT 10h, AH = 02h - Set Cursor Position 5-12
INT 10h, AH = 03h - Read Cursor Position 5-13
INT 10h, AH = 04h - Read Lightpen Position 5-13
INT 10h, AH = 05h - Select Active Display Page 5-14
INT 10h, AH = 06h - Scroll Active Page Up 5-15
INT 10h, AH = 07h - Scroll Active Page Down 5-15
INT 10h, AH = 08h - Read Attribute/Character at Current Cursor Position 5-16
INT 10h, AH = 09h - Write Attribute/Character at Current Cursor Position 5-17
INT 10h, AH = 0Ah - Write Character Only at Current Cursor Position 5-18
INT 10h, AH = 0Bh - Set Color Palette 5-19
INT 10h, AH = 0Ch - Write Pixel 5-20
INT 10h, AH = 0Dh - Read Pixel 5-21
INT 10h, AH = 0Eh - Write TTY Character 5-22
INT 10h, AH = 0Fh - Read Current Video State 5-23
INT 10h, AH = 10h, AL = 00h - Set Individual Palette Register 5-24
INT 10h, AH = 10h, AL = 01h - Set Overscan Color 5-25

5.6 INTERRUPTS (Continued)

INT 10h, AH = 10h, AL = 02h - Set All Palette Registers 5-26
INT 10h, AH = 10h, AL = 03h - Program Blink/Intensity 5-27
INT 10h, AH = 11h, AL = 00h - Load User Text Set 5-28
INT 10h, AH = 11h, AL = 01h - Load ROM Monochrome Set 5-28
INT 10h, AH = 11h, AL = 02h - Load ROM 8 x 8 Double-Dot Set 5-29
INT 10h, AH = 11h, AL = 03h - Set Block Specifier 5-29
INT 10h, AH = 11h, AL = 10h - Load User Text Set 5-30
INT 10h, AH = 11h, AL = 11h - Load ROM Monochrome Set 5-31
INT 10h, AH = 11h, AL = 12h - Load ROM 8 x 8 Double-Dot Set 5-32
INT 10h, AH = 11h, AL = 20h - Load User 8 x 8 Text Set 5-32
INT 10h, AH = 11h, AL = 21h - Load User Graphics Characters 5-33
INT 10h, AH = 11h, AL = 22h - Load ROM 8 x 14 Dot Set 5-33
INT 10h, AH = 11h, AL = 23h - Load ROM 8 x 8 Double-Dot Set 5-34
INT 10h, AH = 11h, AL = 30h - Return Font Information 5-34
INT 10h, AH = 12h, BL = 10h - Return EGA Information 5-35
INT 10h, AH = 12h, BL = 20h - Select Alternate Print Screen Routine 5-38
INT 10h, AH = 13h - Write String 5-38
INT 10h, AH = BFh - Video BIOS Extensions 5-40
INT 10h, AH = BFh, AL = 00h - Switch to External Monitor 5-40
INT 10h, AH = BFh, AL = 01h - Switch to Internal Monitor 5-41
INT 10h, AH = BFh, AL = 02h - Change Control Mode 5-41
INT 10h, AH = BFh, AL = 03h - Read Video Environment 5-42
INT 10h, AH = BFh, AL = 04h - Set Mode-Switch Delay 5-43
INT 1Fh - PTR - Graphics Dot Table Vector 5-44
INT 42h - SW - Compatible Video I/O 5-46
INT 43h - PTR - Graphics Dot Table Vector 5-46
5.7 RAM AREA USED BY THE VIDEO BIOS 5-49

Chapter 5
BIOS

5.1 INTRODUCTION TO VIDEO BIOS

This chapter describes the video BIOS. (It may also be referred to simply as BIOS unless specific reference is made to the system BIOS.)

The video BIOS ROM contained in the COMPAQ Enhanced Color Graphics Board consists of a collection of routines that can be executed by software interrupts (INT). These routines initialize the board and provide a consistent interface to the hardware elements of the system.

The BIOS functions of the board support all interrupts (INT) relating to EGA functions. If a particular BIOS operation is not supported by the COMPAQ Enhanced Color Graphics Board, the board's BIOS issues an INT 42h, which passes control to the system board BIOS routines.

A single 16KB x 8 ROM contains the EGA BIOS, the character font tables, and other initialization parameter tables.

The video BIOS is divided into three sections: initialization, main BIOS, and read-only data. During the power-on ROM scan, the arrangement of the ROM header and verification of its checksum causes the system ROM to do a FAR CALL to offset 0003h in the video BIOS ROM, which is the initialization section.

This initialization section then senses the board and the system configuration, initializes the registers appropriately, and performs a power-on (diagnostic) self-test. When the power-on diagnostic test has successfully completed, a FAR RET instruction causes control to return to the system ROM, which is the main BIOS section.

The main BIOS section provides all INT 10h processing. A function dispatcher fields all INT 10h calls, saves registers, loads common registers, and calls the appropriate handler. When the function code in register AH exceeds 13h, control passes to INT 42h, except for extended functions (AH=BFh). This action provides a chaining process for extending the video BIOS. Individual function code handlers return information, if any, by placing the information in the stack data structure for access.

Over 40 percent of the BIOS consists of read-only data sections, most of which contain dot tables. Because the read-only data sections are accessible by a "public" pointer, they may be copied to RAM and selectively modified as a substitute, or replaced entirely.

5.2 VIDEO BIOS MODES SUPPORTED

The particular BIOS modes supported by the board are a function of the monitor used. Table 5-1 lists the video BIOS modes and the monitors supported by each mode.

Table 5-1. BIOS Modes and Supported Monitors

Video BIOS Mode	Operating Characteristics			Monitors Supported			
	Display	Format	Screen Resolution	COMPAQ Color	COMPAQ Dual-Mode	Enhanced Color	RGBI Color
0	Color	40 x 25 Text	320 x 200	X	X	X	X
1	Color	40 x 25 Text	320 x 200	X	X	X	X
2	Color	80 x 25 Text	640 x 200	X	X	X	X
3	Color	80 x 25 Text	640 x 200	X	X	X	X
4	Color	Graphics	320 x 200	X	X	X	X
5	Color	Graphics	320 x 200	X	X	X	X
6	Color	Graphics	640 x 200	X	X	X	X
7	Monochrome	80 x 25 Text	720 x 350	-	X	-	-
D	Color	Graphics	320 x 200	X	X	X	X
E	Color	Graphics	640 x 200	X	X	X	X

Legend
X = Supported
- = Not supported

(Continued)

Table 5-1. BIOS Modes and Supported Monitors (Concluded)

Video BIOS Mode	Operating Characteristics			Monitors Supported			
	Display	Format	Screen Resolution	COMPAQ Color	COMPAQ Dual-Mode	Enhanced Color	RGBI Color
F (Note)	Monochrome	Graphics	640 x 350	-	X	-	-
10 (Note)	Color	Graphics	640 x 350	X	X	X	-
F+(Note)	Monochrome	Graphics	640 x 350	-	X	-	-
10+(Note)	Color	Graphics	640 x 350	X	X	X	-
0*	Color	40 x 25 Text	320 x 350	X	X	X	-
1*	Color	40 x 25 Text	320 x 350	X	X	X	-
2*	Color	80 x 25 Text	640 x 350	X	X	X	-
3*	Color	80 x 25 Text	640 x 350	X	X	X	-

Legend

X = Supported
- = Not supported

Note: Throughout this chapter, references to BIOS Modes F and 10 are actually references to the full memory implementation of the monochrome graphics Mode F+ and color graphics Mode 10+. Because 256 KB of video RAM is standard on the COMPAQ Enhanced Color Graphics Board, Modes F+ and 10+ are always activated when Modes F and 10 are set, respectively.

5.3 DEFAULT COLOR PALETTE

In the BIOS function call definitions, designations P0..P3 refer to foreground palette selection bits and p0..p3 refer to background palette bits. Actual colors displayed can be arbitrarily assigned by the user. However, default color assignments are as given in Tables 5-2 through 5-6.

Table 5-2. Default Colors for a 4-Color Palette

When Intensity = 0, Palette = 0
Then:

Register	Color
0	Background
1	Green
2	Red
3	Brown

When Intensity = 0, Palette = 1
Then:

Register	Color
0	Background
1	Cyan
2	Magenta
3	White

When Intensity = 1, Palette = 0
Then:

Register	Color
0	Background
1	Light Green
2	Light Red
3	Yellow

When Intensity = 1, Palette = 1
Then:

Register	Color
0	Background
1	Light Cyan
2	Light Magenta
3	Intense White

Table 5-3. Default Colors for a 16-Color Palette

Register	Color
0	Black
1	Blue
2	Green
3	Cyan
4	Red
5	Magenta
6	Brown
7	White
8	Gray
9	Light Blue
10	Light Green
11	Light Cyan
12	Light Red
13	Light Magenta
14	Yellow
15	Intense White

Table 5-4. Default Colors for a 2-Color Palette

Register	Color
0	Black
1	White

Table 5-5. Default Intensities for a 3-Intensity Monochrome Text Display

Register	Intensity
0	Black
1	Normal
2	Normal
3	Highlight

Table 5-6. Default Intensities for a 3-Intensity/Blinking Monochrome Graphics Display

Register	Intensity
0	Black
1	Normal
2	Black
3	Black
4	Highlight to Black Blink
5	Highlight
6	Black
7	Black
8	Black
9	Normal
10	Black
11	Black
12	Black to Highlight Blink
13	Highlight
14	Black
15	Black

5.4 INITIALIZATION

The COMPAQ Enhanced Color Graphics Board BIOS resides in 16 KB of ROM at address C000:0000. However, when the board is added to a COMPAQ DESKPRO 386 Personal Computer, the system BIOS first copies the board BIOS to 32-bit high memory at power-on. This repositioning enables much faster execution speed than can be obtained by access from the 8-bit-wide video BIOS ROM alone. The actual high-memory segment used depends on whether or not other optional system ROMs exist, but the segment typically used is located at FE0000h. Code is written in a position-independent manner so that it can execute without regard to the absolute value in the code segment (CS) register.

5.5 POWER-ON SELF-TEST

During the board power-on self-test, the diagnostic ports at I/O address 0084h and 0085h are used to indicate error status. The video BIOS ROM has a type code of 05; when the self-test process is started, a 05h is output to port 0085h to indicate that the video ROM is conducting a power-on self-test. Port 0084h indicates the specific function currently taking place during the self-test and any errors that might occur.

When an error is detected during the self-test, the system speaker sounds one long beep followed by two short beeps. Beeps are at 1200 hertz (Hz) instead of the typical 900 Hz used by the system board for errors. This beep pattern localizes the error to either the system board or the COMPAQ Enhanced Color Graphics Board. During the power-on self-test process, more than one error can occur, and multiple sets of beeps and error codes may be output.
At the end of the power-on self-test process, both ports 0084h and 0085h are reset to 00h to indicate the end of the self-test process.

5.6 INTERRUPTS

The video BIOS uses five interrupts to perform all video functions. A sixth interrupt (INT 0Ah) is available for the vertical blanking interrupt feature. The interrupts are listed in Table 5-7.

Table 5-7. Video Interrupts

Interrupt	Type	Function
INT 05h	SW	Print Screen
INT 0Ah	HW	Vertical Blanking
INT 10h	SW	Video I/O
INT 1Fh	PTR	Graphics Dot Table Vector
INT 42h	SW	Compatible Video I/O
INT 43h	PTR	Graphics Dot Table Vector

Legend: SW = Software
HW = Hardware
PTR = Pointer

Table 5-8 shows the memory location, size, and function of RAM used by the video interrupts.

Table 5-8. Memory Locations Used by Video Interrupts

Memory Location	Bytes	Function
0000:0014	4	INT 05h Vector
0000:0028	4	INT 0Ah Vector
0000:0040	4	INT 10h Vector
0000:007C	4	INT 1Fh Vector
0000:0108	4	INT 42h Vector
0000:010C	4	INT 43h Vector
0040:0049	1	Current Video Mode
0040:004A	2	Number of Columns on Screen
0040:004C	2	Length of Screen Regeneration Buffer (in bytes)
0040:004E	2	Start of Screen Regeneration Buffer in Video Memory
0040:0050	16	Cursor Save Area for Each Page (8 entries)
0040:0060	2	Cursor Mode (start and end scan line)
0040:0062	1	Current Page Being Displayed
0040:0063	2	Base Address of Active Video Interface Board
0040:0065	1	Current Mode
0040:0066	1	Current Color
0040:0084	1	Number of Character Rows -1
0040:0085	2	Bytes per Character
0040:0087	1	Miscellaneous Information
0040:0088	1	Configuration Bits
0040:00A8	4	Pointer to Table of Save-Area Pointers
0040:0100	1	Print Screen Status

INT 05h - SW - PRINT SCREEN

INT 05h is called to perform a text print screen function using INT 10h function calls for handling the screen and INT 17h function calls for outputting text to the printer. This print screen handler replaces the existing system ROM INT 05h handler when the Alternate Select function (INT 10h, AH=12h, AL=00h, BL=20h) is invoked. It allows for text screens with other than 25 rows by referencing the V_ROWS variable in BIOS RAM.

ROM ACTION: When INT 05h is called, the ROM saves the current cursor location. Address 0040:0100 contains the status of the print screen operation (see INT 0Ah). If the printer is offline or is not ready, the print screen function will timeout in one character time. When printing is complete, the cursor position is restored. Trailing spaces are not compressed. All registers are preserved across the call.

INT 0Ah - HW - IRQ2, VERTICAL BLANKING

A hardware interrupt is provided to allow software to synchronize itself to the beginning of the vertical blanking period. This synchronization is useful for updating controller registers and random portions of the screen so that all changes appear cleanly in the next displayed frame. Common applications include redrawing pointing-device cursors, incrementing controller register contents to effect smooth scrolling and panning, and so on.

The IRQ2 interrupt occurs at a 50- or 60-Hz rate when initiated by writing to bits <5..4> in the Vertical Sync End register of the CRT controller. The interrupt rate and the time allowed for software activity while the screen is blanked depend on the type of monitor connected and the BIOS mode, which are given in Table 5-9.

Table 5-9. Vertical Screen Blanking Time

Monitor Type	BIOS Modes	Rate (Hertz)	Time (Milliseconds)
COMPAQ Color	0..6, D..E	60	3.82
Monitor	10, 0*..3*	60	0.64
COMPAQ	0..6, D..E	60	3.82
Dual-Mode	7, F	50	0.97
Monitor	10, 0*..3*	50	0.97
RGBI Color Monitor	0..6, D..E	60	3.82

On 80286- and 80386-based computers, hardware interrupt IRQ9 is substituted for IRQ2. In this case, the interrupt vectored through INT 71h is redirected by the BIOS to INT 0Ah for system compatibility with 8088- and 8086-based products. Unlike the 8088 and 8086 microprocessors, however, the 80286 and 80386 microprocessors also provide for an INVALID TSS (Task-State Segment) interrupt on INT 0Ah. Therefore, if IRQ9 conflicts with the usage of INT 0Ah, INT 71h may be intercepted directly.

INT 10h - SW - Video I/O

INT 10h is called to perform all video-related BIOS support functions.

ROM ACTION: Upon entry, ROM transfers control to one of 21 routines, given in Table 5-10, based on the function code in register AH. Illegal function codes call the system ROM via INT 42h. All registers that do not return information are preserved.

Table 5-10. INT 10h Video BIOS Calls

INT 10h	Video I/O
AH=00h	Set Mode
AH=01h	Set Cursor Type
AH=02h	Set Cursor Position
AH=03h	Read Cursor Position
AH=04h	Read Lightpen Position
AH=05h	Select Active Display Page
AH=06h	Scroll Active Page Up
AH=07h	Scroll Active Page Down
AH=08h	Read Attribute/Character at Current Cursor Position
AH=09h	Write Attribute/Character at Current Cursor Position
AH=0Ah	Write Character Only at Current Cursor Position
AH=0Bh	Set Color Palette
AH=0Ch	Write Pixel
AH=0Dh	Read Pixel
AH=0Eh	Write TTY Character
AH=0Fh	Read Current Video State
AH=10h	Set Palette Register
AL=00h	Set Individual Palette Register
AL=01h	Set Overscan Color

(Continued)

Table 5-10. INT 10h Video BIOS Calls (Concluded)

AL=02h	Set All Palette Registers
AL=03h	Program Blink/Intensity
AH=11h	Load Character Generator
AL=00h	Load User Text Set
AL=01h	Load ROM Monochrome Set
AL=02h	Load ROM 8 x 8 Double-Dot Set
AL=03h	Set Block Specifier
AL=10h	Load User Text Set
AL=11h	Load ROM Monochrome Set
AL=12h	Load ROM 8 x 8 Double-Dot Set
AL=20h	Load User 8 x 8 Text Set
AL=21h	Load User Graphics Characters
AL=22h	Load ROM 8 x 14 Dot Set
AL=23h	Load ROM 8 x 8 Double-Dot Set
AL=30h	Return Font Information
AH=12h	Alternate Select
BL=10h	Return EGA Information
BL=20h	Select Alternate Print Screen Routine
AH=13h	Write TTY String
AH=BFh	Video BIOS Extensions
AL=00h	Switch to External Monitor
AL=01h	Switch to Internal Monitor
AL=02h	Change Control Mode
AL=03h	Read Video Environment
AL=04h	Set Mode-Switch Delay

Interrupts remain enabled and execution may be suspended if the CTRL + NUMLOCK keys are pressed. Functions and their related parameters are individually described on the following pages.

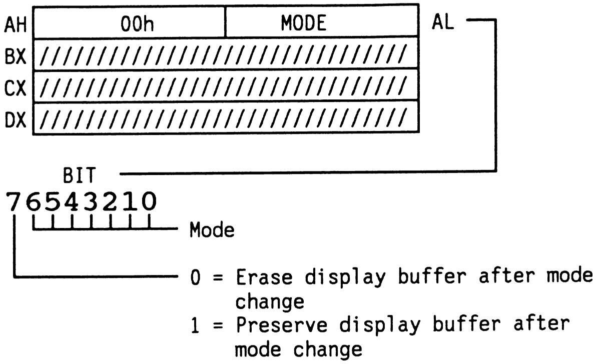
INT 10h, AH = 00h - SET MODE

This function specifies the display mode. It also loads the character sets from the COMPAQ Enhanced Color Graphics Board ROM into the character generator.

Graphics Modes 4, 5, 6, D, E, F, and 10 display characters using ROM firmware to read a look-up table and write the appropriate dots on the screen. INT 1Fh and/or INT 43h point to the dot table(s).

INPUTS:
AH = Function code = 00h
AL = Mode
- 00h Text 40 x 25
- 01h Text 40 x 25
- 02h Text 80 x 25
- 03h Text 80 x 25
- 04h Graphics 320 x 200, 4 Colors
- 05h Graphics 320 x 200, 4 Colors
- 06h Graphics 640 x 200, 2 Colors
- 07h Text 80 x 25, Monochrome
- 0Bh Load Color Character Generator
- 0Ch Load Monochrome Character Generator
- 0Dh Graphics 320 x 200, 16 Colors
- 0Eh Graphics 640 x 200, 16 Colors
- 0Fh Graphics 640 x 350, Monochrome
- 10h Graphics 640 x 350, 16/64 Colors

The difference between two modes that have the same display format is the presence or absence of the color burst signal (used for composite video). Because the COMPAQ Enhanced Color Graphics Board does not have a composite-video output, no difference exists between Modes 0 and 1, 2 and 3, and 4 and 5.


AH
00h
MODE
AL
BX
CX
DX
BIT
76543210
Mode
0 = Erase display buffer after mode change
1 = Preserve display buffer after mode change

OUTPUTS: None.

Mode 7 uses the 9 x 14 dot character cells from the loadable character generator.

INT 10h, AH = 01h - SET CURSOR TYPE

This function establishes the type of cursor to be displayed, if any, in text modes. The shape of the cursor is a function of the illuminated starting and ending scan lines, counting from scan line 0 at the top of the character.

The cursor always blinks at a rate 1/16th of the vertical frame rate. Altering the cursor using bits <7..5> is not compatible with the 6845 CRT controller in other modes and can yield unpredictable results.

INPUTS:
AH = 01h
CH = Start line number for cursor in bits <4..0>
CL = End line number for cursor in bits <4..0>

AH	01h	///////////////	AL
BX	///////////////////////////////		
CH	START LINE NO.	END LINE NO.	CL
DX	///////////////////////////////		

OUTPUTS: None.

INT 10h, AH = 02h - SET CURSOR POSITION

This function sets the cursor position on a given display page according to the row and column specified in register DH.

INPUTS:
AH = 02h
BH = Page number (0..7) for Modes 0..3, 7, D
(0) for Modes 4..6
(0..3) for Mode E
(0..1) for Modes F..10
DH = Row (0..ROWS-1)
DL = Column (0..39) for Modes 0..1, 4..5, D
(0..79) for Modes 2..3, 6..7, E..10

AH	02h	///////////////	AL
BH	PAGE NUMBER	///////////////	BL
CX	///////////////////////////////		
DH	ROW	COLUMN	DL

OUTPUTS: None.

The cursor is not displayable in graphics modes; however, the BIOS does keep track of it in software.

INT 10h, AH = 03h - READ CURSOR POSITION

This function reads the cursor type as defined by its starting and ending scan lines, and the cursor position on the screen as maintained by the BIOS.

INPUTS:
AH = 03h
BH = Page number (0..7) for Modes 0..3, 7, D
(0) for Modes 4..6
(0..3) for Mode E
(0..1) for Modes F..10

AH	03h	///////////////	AL
BH	PAGE NUMBER	///////////////	BL
CX	////////////////////////////////		
DX	////////////////////////////////		

OUTPUTS:
CH = Start scan-line number for cursor in bits <4..0>
CL = End scan-line number for cursor in bits <4..0>
DH = Row
DL = Column

AX	////////////////////////////////		
BX	////////////////////////////////		
CH	START LINE NO.	END LINE NO.	CL
DH	ROW	COLUMN	DL

INT 10h, AH = 04h - READ LIGHTPEN POSITION

This function reads the status and position of an optional lightpen and returns its position in pixel coordinates.

INPUTS: AH = 04h

AH	04h	///////////////	AL
BX	////////////////////////////////		
CX	////////////////////////////////		
DX	////////////////////////////////		

OUTPUTS:
AH = 0 Lightpen switch not down (not triggered)
= 1 Valid lightpen in registers
BX = Pixel column (0..319) for Modes 4..5, D
(0..639) for Modes 6, E..10
CH = Raster line (0..199) for Modes 4..6
CX = Raster line (0..199) for Modes D..E
(0..349) for Modes F..10
DH = Character row lightpen is on
DL = Character column lightpen is on

AH	PEN VALID	///////////////	AL	
BX	PIXEL COLUMN			
CH	RASTER LINE	///////////////	CL	(See Note)
CX	RASTER LINE			
DH	CHAR. ROW	CHAR. COLUMN	DL	

Note: The raster line is returned in CH for Modes 4, 5, and 6. It is returned in CX for Modes D, E, F, and 10.

The lightpen resolution is equivalent to the character-grid layout.

Although the BIOS software and interface hardware support the use of the lightpen in Modes 7 and F, the monitor in use may not.

INT 10h, AH = 05h - SELECT ACTIVE DISPLAY PAGE

This function selects the active display page. The number of pages available is a function of the mode.

INPUTS:
AH = 05h
AL = New page value
    (0..7) for Modes 0..3, 7, D
    (0..3) for Mode E
    (0..1) for Modes F..10

AH	05h	NEW PAGE VALUE	AL
BX	/////////////////////////////////		
CX	/////////////////////////////////		
DX	/////////////////////////////////		

OUTPUTS: None

INT 10h, AH = 06h - SCROLL ACTIVE PAGE UP

This function scrolls text up within an arbitrarily defined window from a size of one character to the entire screen. This same function call can also be used to blank a window. See Section 5.3, Default Color Palette, for a definition of the attribute byte in register BH.

INPUTS:
AH = 06h
AL = Number of lines to scroll (0..ROWS) [See Note]
(AL = 0 means blank entire window)
BH = Attribute to be used on blank lines (00h..FFh)
[Mode dependent]
CH = Row of upper-left corner of scroll (0..DH)
CL = Column of upper-left corner of scroll (0..DL)
DH = Row of lower-right corner of scroll (CH..ROWS-1)
DL = Column of lower-right corner of scroll
(CL..39) for Modes 0..1, 4..5, D
(CL..79) for Modes 2..3, 7, E..10

AH	06h	NUMBER OF LINES	AL
BH	ATTRIBUTE	//////////////	BL
CH	U.L. ROW	U.L. COLUMN	CL
DH	L.R. ROW	L.R. COLUMN	DL

OUTPUTS:None.

Note: Clipping is performed if lower-right (L.R.) COLUMN value is greater than WIDTH - 1.

INT 10h, AH = 07h - SCROLL ACTIVE PAGE DOWN

This function scrolls text down within an arbitrarily defined window from a size of one character to the entire screen. This same function call can also be used to blank a window. See Section 5.3, Default Color Palette, for a definition of the attribute byte in register BH.

INPUTS:
AH = 07h
AL = Number of lines to scroll (0..ROWS) [See Note]
(AL = 0 means blank entire window)
BH = Attribute to be used on blank lines (00h..FFh)
[Mode dependent]
CH = Row of upper-left corner of scroll (0..DH)
CL = Column of upper-left corner of scroll (0..DL)
DH = Row of lower-right corner of scroll (CH..ROWS-1)
DL = Column of lower-right corner of scroll
(CL..39) for Modes 0..1, 4..5, D
(CL..79) for Modes 2..3, 7, E..10

AH	07h	NUMBER OF LINES	AL
BH	ATTRIBUTE	//////////////	BL
CH	U.L. ROW	U.L. COLUMN	CL
DH	L.R. ROW	L.R. COLUMN	DL

OUTPUTS:None.

Note: Clipping is performed if L.R. COLUMN value is greater than WIDTH - 1.

INT 10h, AH = 08h - READ ATTRIBUTE/CHARACTER AT CURRENT CURSOR POSITION

This function reads the character and its attribute at the current cursor position on a given display page.

INPUTS:
AH = 08h
BH = Page number (0..7) for Modes 0..3, 7, D
(0) for Modes 4..6
(0..3) for Mode E
(0..1) for Modes F..10

AH	08h	///////////////	AL
BH	DISPLAY PAGE	///////////////	BL
CX	//////////////////////////////////		
DX	//////////////////////////////////		

OUTPUTS:
AL = Character read
AH = Attribute of character read

AH	ATTRIBUTE	CHAR. READ	AL
BX	//////////////////////////////////		
CX	//////////////////////////////////		
DX	//////////////////////////////////		

For read and write character functions while in graphics mode, the characters are formed by the BIOS from a dot table. Interrupt vectors INT 1Fh and/or INT 43h point to the table(s).

For Modes 4..6 and D..10 (graphics modes), a space can be written with a 00h or a 20h, but only a 00h is read back.

For Modes 4..6 and D..10, the attribute returned in AH is not valid.

INT 10h, AH = 09h - WRITE ATTRIBUTE/CHARACTER AT CURRENT CURSOR POSITION

This function writes one or more repeated characters with attributes on a given display page at the current cursor position. (For a definition of the color associated with the attribute byte in BL, see Section 5.3, Default Color Palette.)

INPUTS:
AH = 09h
AL = Character to write (00h..FFh)
BH = Page number (0..7) for Modes 0..3, 7, D
(0) for Modes 4..6
(0..3) for Mode E
(0..1) for Modes F..10
BL = Attribute of character to write
In graphics modes only:
If bit 7 of BL = 0, then the character pattern denoted by the byte in AL replaces the contents of the character cell.
If bit 7 of BL = 1, then the character pattern denoted by the byte in AL is XORed with the contents of the character cell.
CX = Count of characters to write
(1..1024) for Modes 0..1
(1..2048) for Modes 2..3, 7
(1..40) for Modes 4..5, D
(1..80) for Modes 6, E..10

AH	09h	CHARACTER	AL
BH	DISPLAY PAGE	ATTRIBUTE	BL
CX	COUNT OF CHARACTERS TO WRITE		
DX	//////////////////////////////////		

OUTPUTS: None.

For read and write character functions while in graphics mode, the characters are formed by the BIOS from a dot table. Interrupt vectors INT 1Fh and/or INT 43h point to the table(s).

The maximum character count is limited in the text modes to the end of the display page.

When in Modes 4..6 or D..10 (graphics modes), the replication factor contained in register CX produces valid results only for characters on the same row.

INT 10h, AH = 0Ah - WRITE CHARACTER ONLY AT CURRENT CURSOR POSITION

This function writes one or more repeated characters without affecting the attribute on a given display page at the current cursor position, without moving the cursor.

INPUTS:
AH = 0Ah
AL = Character to write (00h..FFh)
BH = Page number (0..7) for Modes 0..3, 7, D
(0) for Modes 4..6
(0..3) for Mode E
(0..1) for Modes F..10
BL = In text modes: ignored.
In graphics modes: attribute of character to write. If bit 7 of BL = 0, then the character pattern denoted by the byte in AL replaces the contents of the character cell.
If bit 7 of BL = 1, then the character pattern denoted by the byte in AL is XORed with the contents of the character cell.
CX = Count of characters to write
(1..1024) for Modes 0..1
(1..2048) for Modes 2..3, 7
(1..40) for Modes 4..5, D
(1..80) for Modes 6, E..10

AH	09h	CHARACTER	AL
BH	DISPLAY PAGE	ATTRIBUTE	BL
CX	COUNT OF CHARACTERS TO WRITE		
DX	////////////////////////////////		

OUTPUTS: None

For read and write character functions while in graphics mode, the characters are formed by the BIOS from a dot table. Interrupt vectors INT 1Fh and/or INT 43h point to the table(s).

The maximum character count is limited in the text modes to the end of the display page.

When in Modes 4..6 or D..10 (graphics modes), the maximum character count contained in register CX produces valid results only for characters on the same row.

When in Modes 4..6 or D..10, BL=ATTRIBUTE; otherwise, BL is ignored.

INT 10h, AH = 0Bh - SET COLOR PALETTE

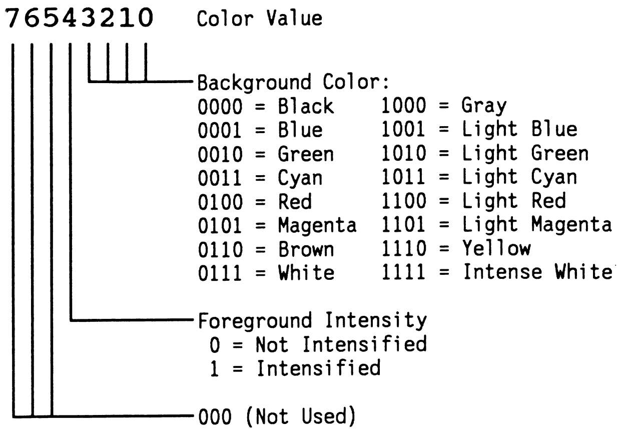
This function selects the background color, foreground intensity, and one of two color palettes for Modes 4 and 5. In Mode 6, this function selects the foreground color. The background color in Mode 6 is always black.

INPUTS:
AH = 0Bh
BH = Palette color ID being set (0..127)
BL = Color value to be used with that color ID.

AH	0Bh	///////////////	AL
BH	COLOR ID	COLOR VALUE	BL
CX	////////////////////////////////		
DX	////////////////////////////////		

For Modes 4 and 5:

Color ID		Function
0	=	Set Background/Intensity (0..31)



76543210 Color Value

Background Color:
0000 = Black
0001 = Blue
0010 = Green
0011 = Cyan
0100 = Red
0101 = Magenta
0110 = Brown
0111 = White
1000 = Gray
1001 = Light Blue
1010 = Light Green
1011 = Light Cyan
1100 = Light Red
1101 = Light Magenta
1110 = Yellow
1111 = Intense White

Foreground Intensity
0 = Not Intensified
1 = Intensified

000 (Not Used)

For Mode 6:

Color ID		Function
0	=	Set Foreground (0..15)

76543210 Color Value

Foreground Color:

0000 = Black
0001 = Blue
0010 = Green
0011 = Cyan
0100 = Red
0101 = Magenta
0110 = Brown
0111 = White
1000 = Gray
1001 = Light Blue
1010 = Light Green
1011 = Light Cyan
1100 = Light Red
1101 = Light Magenta
1110 = Yellow
1111 = Intense White

0000 (Not Used)

The color values listed reflect default settings only. The actual color rendered on the screen is a function of palette programming. (Refer to Section 5.3, Default Color Palette.)

INT 10h, AH = 0Ch - WRITE PIXEL

This function writes a single pixel in the graphics modes.

INPUTS:

AH = 0Ch

AL = Color value (see Section 5.3, Default Color Palette.)
    (00h..03h) or (80h..83h) for Modes 4..5
    (00h..01h) or (80h..81h) for Mode 6
    (00h..0Fh) or (80h..8Fh) for Modes D..10
    If bit 7 of AL = 0, then the color value replaces the current contents of the pixel.
    If bit 7 of AL = 1, then the color value is XORed with the current contents of the pixel.

BH = Page number (0) for Modes 4..6
    (0..7) for Mode D
    (0..3) for Mode E
    (0..1) for Modes F..10

CX = Column number
    (0..319) for Modes 4..5, D
    (0..639) for Mode 6, E..10

DX = Row number
    (0..199) for Modes 4..6, D..E
    (0..349) for Modes F..10

AH	0Ch	COLOR VALUE	AL
BH	DISPLAY PAGE	///////////////	BL
CX	COLUMN NUMBER		
DX	ROW NUMBER		

OUTPUTS: None.

This function has significance only for graphics modes.

INT 10h, AH = 0Dh - READ PIXEL

This function reads a single pixel in graphics mode.

INPUTS:
AH = 0Dh
BH = Page number (0) for Modes 4..6
(0..7) for Mode D
(0..3) for Mode E
(0..1) for Modes F..10
CX = Column number
(0..319) for Modes 4..5, D
(0..639) for Modes 6, E..10
DX = Row number
(0..199) for Modes 4..6, D..E
(0..349) for Modes F..10

AH	0Dh	///////////////	AL
BH	DISPLAY PAGE	///////////////	BL
CX	COLUMN NUMBER		
DX	ROW NUMBER		

OUTPUTS:
AL = Color value
(00h..03h) for Modes 4..5
(00h..01h) for Mode 6
(00h..0Fh) for Modes D..10

AH	///////////////	DOT VALUE	AL
BX	/////////////////////////////////		
CX	/////////////////////////////////		
DX	/////////////////////////////////		

This function has significance only for the graphics modes.

INT 10h, AH = 0Eh - WRITE TTY CHARACTER

This function writes a character to the active display page and executes the traditional carriage-control characters: CR, LF, BS, and BEL, moving the cursor after writing the character. When the rightmost column of the screen is written, the cursor effectively does a CR, LF sequence.

Carriage-Control Character Functions:	
CR	The carriage-return character (0Dh) moves the cursor to column 0 of the current line.
LF	The line-feed character (0Ah) moves the cursor down one row on the screen, provided the present row is not the bottom row. If the cursor is already on the bottom row, the screen is scrolled up using the attribute of the leftmost character in column 0 of the bottom row for the attribute of the new line.
BS	The backspace character (08h) moves the cursor one position to the left. If it is already in column 0, nothing happens.
BEL	The bell character (07h) causes a 900-Hz tone to sound for approximately 500 milliseconds (ms).

INPUTS:
AH = 0Eh
AL = Character to write (00h..FFh)
BL = Foreground color in graphics mode
(00h..03h) for Modes 4..5
(00h..01h) for Mode 6
(00h..0Fh) for Modes D..10

AH	0Eh	CHARACTER	AL
BH	///////////////	FOREGND COLOR	BL
CX	////////////////////////////////		
DX	////////////////////////////////		

OUTPUTS: None.

Writes are performed only to the active page as defined by the contents of the location at 0040:0062.

Screen width is controlled by the previously set mode.

INT 10h, AH = 0Fh - READ CURRENT VIDEO STATE

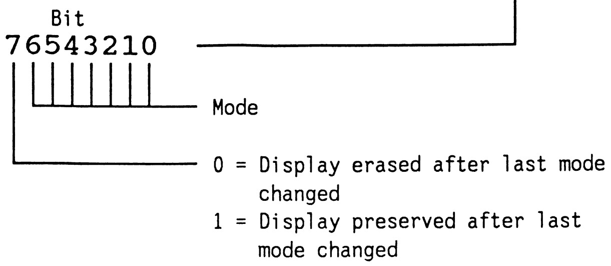
INT 10h, AH = 0Fh returns the current video state.

INPUTS: AH = 0Fh

AH	0Fh	///////////////	AL
BX	////////////////////////////////		
CX	////////////////////////////////		
DX	////////////////////////////////		

OUTPUTS:
AH = Number of character columns on screen (40 or 80)
AL <6..0> = Mode currently set (00h..10h)
<7> = Screen Memory Not Cleared Flag
BH = Current active display page (0..7)

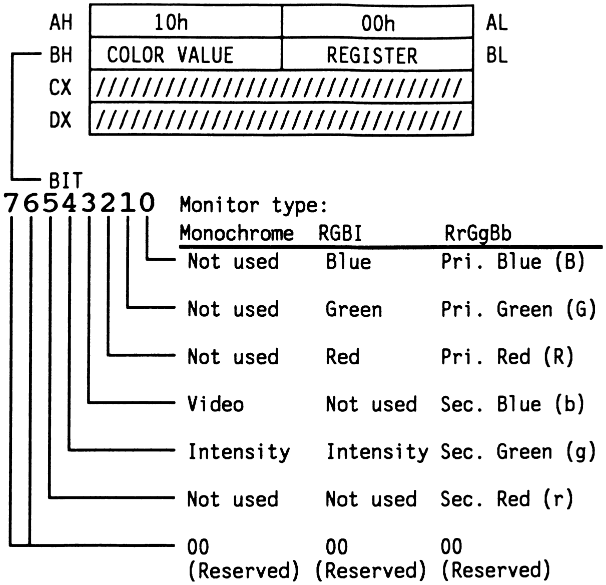
AH	SCREEN WIDTH	CURRENT MODE	AL
BH	ACTIVE PAGE	///////////////	BL
CX	////////////////////////////////		
DX	////////////////////////////////		


Bit
76543210
Mode
0 = Display erased after last mode changed
1 = Display preserved after last mode changed

INT 10h, AH = 10h, AL = 00h - SET INDIVIDUAL PALETTE REGISTER

This function sets an individual palette register or other special-purpose register in the attribute controller.

INPUTS:
AH = 10h
AL = 00h
BH = Color value (00h..3Fh)
The interpretation of the color value depends on the type of monitor in use (see the following bit map).
BL = Register (00h..13h)



AH	10h	00h	AL
BH	COLOR VALUE	REGISTER	BL
CX	////////////////////////////////		
DX	////////////////////////////////		

Monitor type:

BIT	Monochrome	RGBI	RrGgBb
0	Not used	Blue	Pri. Blue (B)
1	Not used	Green	Pri. Green (G)
2	Not used	Red	Pri. Red (R)
3	Video	Not used	Sec. Blue (b)
4	Intensity	Intensity	Sec. Green (g)
5	Not used	Not used	Sec. Red (r)
7 6	00 (Reserved)	00 (Reserved)	00 (Reserved)

OUTPUTS: None.

Palette register 11h is the overscan (border) color.

INT 10h, AH = 10h, AL = 01h - SET OVERSCAN COLOR

This function sets the color of the overscan (border) area on the screen.

INPUTS:
AH = 10h
AL = 01h
BH = Color value (00h..3Fh)
The interpretation of the color value depends on the type of monitor in use (see the following bit map).

AH	10h	01h	AL
BH	COLOR VALUE	///////////////	BL
CX	//////////////////////////////////		
DX	//////////////////////////////////		

BH BIT 76543210 Monitor type:

BIT	Monochrome	RGBI	RrGgBb
0	Not used	Blue	Pri. Blue (B)
1	Not used	Green	Pri. Green (G)
2	Not used	Red	Pri. Red (R)
3	Video	Not used	Sec. Blue (b)
4	Intensity	Intensity	Sec. Green (g)
5	Not used	Not used	Sec. Red (r)
7, 6	00 (Reserved)	00 (Reserved)	00 (Reserved)

OUTPUTS: None.

This function should be used only in 200-scan line formats (Modes 0..6, D..E), because in 350-scan line modes (Modes 0*..3*, 7, F..10) almost no overscan area is available for the border.

INT 10h, AH = 10h, AL = 02h - SET ALL PALETTE REGISTERS

This function sets all color palette registers (including the border) at once.

INPUTS:
AH = 10h
AL = 02h
ES:DX = Pointer to list

AH	10h	02h	AL
BX	///////////////////////////////		
CX	///////////////////////////////		
DX	LIST OFFSET		
ES	LIST SEGMENT		

List format for 200-scan line color operation (Modes 0..6, D..E):

Register	7	6	5	4	3	2	1	0	Offset
Palette 0	0	0	0	I	0	R	G	B	+00h
1	0	0	0	I	0	R	G	B	+01h
.									
.									
Palette 15	0	0	0	I	0	R	G	B	+0Fh
OVERSCAN	0	0	0	I	0	R	G	B	+10h

List format for 350-scan line color operation (Modes 10, 0*..3*):

Register	7	6	5	4	3	2	1	0	Offset
Palette 0	0	0	r	g	b	R	G	B	+00h
1	0	0	r	g	b	R	G	B	+01h
.									
.									
Palette 15	0	0	r	g	b	R	G	B	+0Fh
OVERSCAN	0	0	0	0	0	0	0	0	+10h

List format for 350-scan line monochrome operation (Modes 7, F):

Register	7	6	5	4	3	2	1	0	Offset
Palette 0	0	0	0	I	V	0	0	0	+00h
1	0	0	0	I	V	0	0	0	+01h
.									
.									
Palette 15	0	0	0	I	V	0	0	0	+0Fh
OVERSCAN	0	0	0	0	0	0	0	0	+10h

OUTPUTS: None.

INT 10h, AH = 10h, AL = 03h - PROGRAM BLINK/INTENSITY

This function defines the interpretation of bit <7> in the text-mode attribute bytes or of bit <3> in the graphics mode pixel data. In enhanced graphics modes (D..10), pixel values are affected when the mode in register BL is set to a 1 (blink). Pixel values in the range 0..7 assume the colors in palette registers 0..7. Pixel values in the range 8..15 have colors that toggle (blink) between the colors in palette registers 0..7 and 8..15, respectively.

INPUTS:
AH = 10h
AL = 03h
BL = Mode (0..1)
    0 = Intensify
    1 = Blink

AH	10h	03h	AL
BH	///////////////	MODE	BL
CX	////////////////////////////////		
DX	////////////////////////////////		

OUTPUTS: None.

INT 10h, AH = 11h, AL = 00h - LOAD USER TEXT SET

This function loads a user-defined text mode character set. It initiates a Set Mode without clearing the display buffer.

INPUTS:
AH = 11h
AL = 00h
BH = Bytes per char (1..32)
BL = Block to load (0..3)
CX = Character count
DX = Character offset into table
ES:BP = Pointer to user table

AH	11h	00h	AL
BH	NUMBER OF BYTES	BLOCK TO LOAD	BL
CX	CHARACTER COUNT		
DX	CHARACTER OFFSET INTO TABLE		

BP	TABLE OFFSET

ES	TABLE SEGMENT

OUTPUTS: None.

INT 10h, AH = 11h, AL = 01h - LOAD ROM MONOCHROME SET

This function loads a ROM-resident text mode character set for the monochrome (9 x 14 text font) display.

INPUTS:
AH = 11h
AL = 01h
BL = Block to load (0..3)

AH	11h	01h	AL
BH	//////////////	BLOCK TO LOAD	BL
CX	////////////////////////////////		
DX	////////////////////////////////		

OUTPUTS: None.

INT 10h, AH = 11h, AL = 02h - LOAD ROM 8 x 8 DOUBLE-DOT SET

This function loads a ROM-resident text mode character set for the graphics (8 x 8-dot) display.

INPUTS:
AH = 11h
AL = 02h
BL = Block to load (0..3)

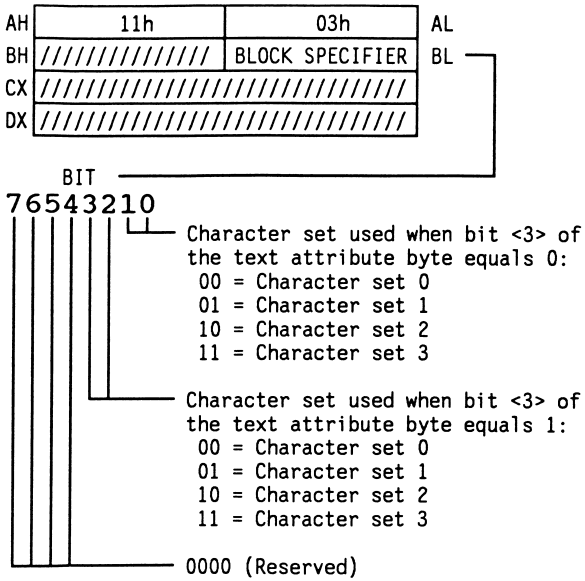
AH	11h	02h	AL
BH	///////////////	BLOCK TO LOAD	BL
CX	////////////////////////////////		
DX	////////////////////////////////		

OUTPUTS: None.

INT 10h, AH = 11h, AL = 03h - SET BLOCK SPECIFIER

This function programs which of three character tables (blocks) to use when the text attribute bit <3> = 0, and which of three character tables to use when the text attribute bit <3> = 1.

INPUTS:
AH = 11h
AL = 03h
BL = Block specifiers (00h..0Fh)



Character set used when bit <3> of the text attribute byte equals 0:
00 = Character set 0
01 = Character set 1
10 = Character set 2
11 = Character set 3

Character set used when bit <3> of the text attribute byte equals 1:
00 = Character set 0
01 = Character set 1
10 = Character set 2
11 = Character set 3

0000 (Reserved)

OUTPUTS: None.

It is recommended that INT 10h, function AH = 10h (Set Palette Registers), subfunction 00h (Set Individual Register) be called first with BH = 07h and BL = 12h. (Ignore attribute bit <3> when generating the color outputs; this generates consistent color outputs for both character sets.)

INT 10h, AH = 11h, AL = 10h - LOAD USER TEXT SET

This function loads a user-defined text mode character set. The number of character rows displayed is dependent on the current display. The subfunctions 10, 11, and 12 are identical to subfunctions 0, 1, and 2. The following CRT controller register values are recalculated as follows:

Index	Calculated Value
07h	Overflow
09h	Maximum Scan Line = bytes_per_character - 1
0Ah	Cursor Start = bytes_per_character - 2
0Bh	Cursor End = 0
12h	Vertical Display End = [(rows + 1) x bytes_per_character] - 1

INPUTS: AH = 11h
AL = 10h
BH = Bytes per character (1..32)
BL = Block to load (0..3)
CX = Character count
DX = Character offset into table
ES:BP = Pointer to user table

AH	11h	10h	AL
BH	NUMBER OF BYTES	BLOCK TO LOAD	BL
CX	CHARACTER COUNT		
DX	CHARACTER OFFSET INTO TABLE		

BP	TABLE OFFSET

ES	TABLE SEGMENT

OUTPUTS: None.

The active page must be Page 0.

INT 10h, AH = 11h, AL = 11h - LOAD ROM MONOCHROME SET

This function loads a ROM-resident text mode character set for the monochrome (9 x 14) character set.

INPUTS:
AH = 11h
AL = 11h
BL = Block to load (0..3)

AH	11h	11h	AL
BH	///////////////	BLOCK TO LOAD	BL
CX	////////////////////////////////		
DX	////////////////////////////////		

OUTPUTS: None.

The active page must be Page 0.

INT 10h, AH = 11h, AL = 12h - LOAD ROM 8 x 8 DOUBLE-DOT SET

This function loads a ROM-resident text mode character set for the graphics 8 x 8 character set display.

INPUTS:
AH = 11h
AL = 12h
BL = Block to load (0..3)

AH	11h	12h	AL
BH	///////////////	BLOCK TO LOAD	BL
CX	//////////////////////////////////		
DX	//////////////////////////////////		

OUTPUTS: None.

The active page must be Page 0.

INT 10h, AH = 11h, AL = 20h - LOAD USER 8 x 8 TEXT SET

This function loads a user-defined text mode character set. It is meant to be called immediately after a Set Mode function is performed. The upper-128 characters of the user 8 x 8 character table are pointed to by INT 1Fh.

INPUTS:
AH = 11h
AL = 20h
ES:BP = Pointer to user table

AH	11h	20h	AL
BX	//////////////////////////////////		
CX	//////////////////////////////////		
DX	//////////////////////////////////		

BP	TABLE OFFSET

ES	TABLE SEGMENT

OUTPUTS: None.

INT 10h, AH = 11h, AL = 21h - LOAD USER GRAPHICS CHARACTERS

This function loads a user-defined graphics character set.

INPUTS:
AH = 11h
AL = 21h
BL = Rows specifier (0..3)
    0 = User set (DL = No. of rows)
    1 = 14 rows
    2 = 25 rows
    3 = 43 rows
CX = Bytes per character
DL = Number of rows (for BL = 0 only)

AH	11h	21h	AL
BH	////////////////	ROWS SPECIFIER	BL
CX	BYTES PER CHARACTER		
DH	////////////////	NO. OF ROWS	DL

BP	TABLE OFFSET

ES	TABLE SEGMENT

OUTPUTS: None.

INT 10h, AH = 11h, AL = 22h - LOAD ROM 8 x 14 DOT SET

This function loads a ROM-resident 8 x 14-dot character font.

INPUTS:
AH = 11h
AL = 22h
BL = Rows specifier (0..3)
    0 = User set (DL = No. of rows)
    1 = 14 rows
    2 = 25 rows
    3 = 43 rows
DL = Number of rows (For BL = 0 only)

AH	11h	22h	AL
BH	////////////////	ROWS SPECIFIER	BL
CX	////////////////////////////////		
DH	////////////////	NO. OF ROWS	DL

OUTPUTS: None.

INT 10h, AH = 11h, AL = 23h - LOAD ROM 8 x 8 DOUBLE-DOT SET

This function loads a ROM-resident 8 x 8 double-dot character font.

INPUTS:
AH = 11h
AL = 23h
BL = Rows specifier (0..3)
- 0 = User set (DL = No. of rows)
- 1 = 14 rows
- 2 = 25 rows
- 3 = 43 rows

DL = Number of rows (For BL = 0 only)

AH	11h	23h	AL
BH	////////////////	ROWS SPECIFIER	BL
CX	/////////////////////////////////		
DH	////////////////	NO. OF ROWS	DL

OUTPUTS:None.

INT 10h, AH = 11h, AL = 30h - RETURN FONT INFORMATION

This function returns information related to the current character font.

INPUTS:
AH = 11h
AL = 30h
BH = Pointer specifier (0..5)
- 0 = INT 1Fh
- 1 = INT 43h
- 2 = ROM 8 x 14 font
- 3 = ROM 8 x 8 double-dot font
- 4 = ROM 8 x 8 upper-128 double-dot font
- 5 = ROM 9 x 14 alternate text font

AH	11h	30h	AL
BH	PTR SPECIFIER	////////////////	BL
CX	/////////////////////////////////		
DX	/////////////////////////////////		

OUTPUTS:
CX = Bytes per character
DL = Character rows on screen
ES:BP = Specified pointer

AX	//////////////////////////////////		
BX	//////////////////////////////////		
CX	BYTES PER CHARACTER		
DH	////////////////	CHAR. ROWS	DL

BP	POINTER OFFSET

ES	POINTER SEGMENT

INT 10h, AH = 12h, BL = 10h - RETURN EGA INFORMATION

This function retrieves information about the configuration and operating mode.

INPUTS:
AH = 12h
AL = 00h
BL = 10h

AH	12h	00h	AL
BH	////////////////	10h	BL
CX	//////////////////////////////////		
DX	//////////////////////////////////		

OUTPUTS:
BH = Video controller mode (0..1)
- 0 = Color (3D4h)
- 1 = Monochrome (3B4h)

BL = Memory size (0..3) [see Note 1]
- 0 = 64 KB
- 1 = 128 KB
- 2 = 192 KB
- 3 = 256 KB (Std)

CH = Feature connector bits [see Note 2 and Figure 5-1]
CL = Video configuration code (00h..0Fh) [see Table 5-11]

AX	////////////////////////////////		
BH	EMULATION	MEMORY SIZE	BL
CH	FEATURE BITS	CONFIG. CODE	CL
DX	////////////////////////////////		

Notes:

1. Only the 256-KB memory configuration is supported. Other values are shown for completeness and reflect alternative configurations that are possible on some compatible enhanced graphics adapters.

2. The feature bits returned by this BIOS call reflect their state as of power-on. This function does not return the real-time status. Any changes since power-on are not reflected. For live status, it is necessary to read port 3C2h, bits <6..5>. See Figure 5-1.

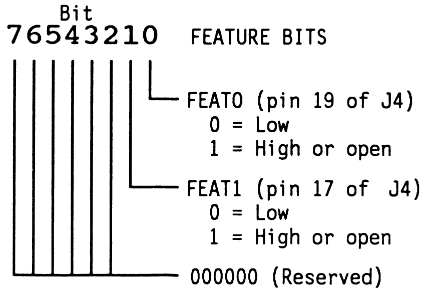

Bit
76543210 FEATURE BITS
FEAT0 (pin 19 of J4)
0 = Low
1 = High or open
FEAT1 (pin 17 of J4)
0 = Low
1 = High or open
000000 (Reserved)

Figure 5-1. Illustration of the Feature Bits Register

Table 5-11 lists the codes returned in register CL that correspond to SW1 settings and the current video configuration.

Table 5-11. CL Register Codes and Their Corresponding Video Configurations

CL Register	Primary Display				Secondary Display (Optional)			
Code	Adapter	Monitor	Text	Resolution	Adapter	Monitor	Text	Resolution
00h	MDA	Mono.	80 x 25	720 x 350	CECG	RGBI	40 x 25	320 x 200
01h	MDA	Mono.	80 x 25	720 x 350	CECG	RGBI	80 x 25	640 x 200
02h	MDA	Mono.	80 x 25	720 x 350	CECG	RrGgBb	80 x 25	640 x 200
03h	MDA	Mono.	80 x 25	720 x 350	CECG	RrGgBb	80 x 25	640 x 350
04h	CGA	RGBI	40 x 25	320 x 200	CECG	DM	80 x 25	720 x 350
05h	CGA	RGBI	80 x 25	640 x 200	CECG	DM	80 x 25	720 x 350
06h	CECG	RGBI	40 x 25	320 x 200	MDA	Mono.	80 x 25	720 x 350
07h	CECG	RGBI	80 x 25	640 x 200	MDA	Mono.	80 x 25	720 x 350
08h	CECG	RrGgBb	80 x 25	640 x 200	MDA	Mono.	80 x 25	720 x 350
09h	CECG	RrGgBb	80 x 25	640 x 350	MDA	Mono.	80 x 25	720 x 350
0Ah	CECG	DM	80 x 25	720 x 350	CGA	RGBI	40 x 25	320 x 200
0Bh	CECG	DM	80 x 25	720 x 350	CGA	RGBI	80 x 25	640 x 200
0Ch : : 0Fh	[Reserved] : [Reserved]				[Reserved] : [Reserved]			

Legend:

MDA = Monochrome display adapter or compatible
Mono. = Compatible monochrome monitor
CECG = COMPAQ Enhanced Color Graphics Board
RGBI = Standard RGBI color monitor
RrGgBb = COMPAQ Color Monitor or compatible enhanced color monitor
CGA = Color graphics adapter or COMPAQ Video Display Controller Board
DM = COMPAQ Dual-Mode Monitor

INT 10h, AH = 12h, BL = 20h - SELECT ALTERNATE PRINT SCREEN ROUTINE

This function sets a new text Print Screen routine (INT 05h) that adjusts for a variable number of rows on the screen other than just the normal 25 rows.

INPUTS:
AH = 12h
AL = 00h
BL = 20h

AH	12h	00h	AL
BH	///////////////	20h	BL
CX	//////////////////////////////////		
DX	//////////////////////////////////		

OUTPUTS: None.

INT 10h, AH = 13h - WRITE STRING

INT 10h, AH = 13h writes ASCII character strings to the currently active display.

INPUTS:
AH = 13h
AL = Format code (0..3)
    0 = format of string is: {char, char, ... , char} and the cursor does NOT move
    1 = format of string is: {char, char, ... , char} and the cursor DOES move
    2 = format of string is: {char, attr, char, attr, ... ,char, attr} and the cursor does NOT move (see Note)
    3 = format of string is: {char, attr, char, attr, ... ,char, attr} and the cursor DOES move (see Note)
BH = Page number (0..7) for Modes 0..3, 7, D
                 (0) for Modes 4..6
                 (0..3) for Mode E
                 (0..1) for Modes F..10
BL = Attribute (00h..FFh)
CX = Length of string (Text = 0..screen_size)
                      (Graphics = 0..end_of_line)
DH = Cursor row position (0..rows-1)
DL = Cursor column position (0..width-1)
ES:BP = Pointer to string

AH	13h	FORMAT CODE	AL
BH	DISPLAY PAGE	ATTRIBUTE	BL
CX	LENGTH OF STRING		
DH	CURSOR ROW	CURSOR COLUMN	DL

BP	STRING OFFSET

ES	STRING SEGMENT

OUTPUTS: None.

Note: Imbedded carriage control characters CR, LF, BS, and BEL do not have an associated attribute byte for AL=2 or 3; the next character immediately follows.

The video Write String function always scrolls the active page even when writing to pages other than the active page. This occurs because a Write TTY Character is used with a LF to perform the scroll. On 80286-based products, the Write TTY Character function ignores the page specified in BH and always writes to the active page. Carriage-control characters CR, LF, BS, and BEL are also written using Write TTY Character and can cause a change in cursor position on the active page. When the screen scrolls, the attribute used for the new line is the same as the attribute for the rightmost character on the bottom row.

INT 10h, AH = BFh - VIDEO BIOS EXTENSIONS

Function BFh adds extensions to the video BIOS needed for sensing or altering the hardware environment.

INPUTS:
AH = BFh
AL = 00h = Switch to External Monitor
     01h = Switch to Internal Monitor
     02h = Set Control Mode
     03h = Get Environment
     04h = Set Mode-Switch Delay
     05h = [Reserved]
       :
     FFh = [Reserved]

AH	BFh	SUBFUNCTION	AL
BX	//////////////////////////////////		
CX	//////////////////////////////////		
DX	//////////////////////////////////		

INT 10h, AH = BFh, AL = 00h - SWITCH TO EXTERNAL MONITOR

This interrupt directs the BIOS to switch to the external monitor. If the monitor is not present or cannot support the format, then nothing happens and control returns to the caller. This BIOS function facilitates the CTRL + ALT + < feature.

INPUTS:
AH = BFh
AL = 00h

AH	BFh	00h	AL
BX	//////////////////////////////////		
CX	//////////////////////////////////		
DX	//////////////////////////////////		

OUTPUTS: None

INT 10h, AH = BFh, AL = 01h - SWITCH TO INTERNAL MONITOR

This interrupt directs the BIOS to switch to the internal monitor. If the monitor is not present or cannot support the format, then nothing happens and control returns to the caller. This BIOS function supports the CTRL + ALT + > feature.

INPUTS:
AH = BFh
AL = 01h

AH	BFh	01h	AL
BX	//////////////////////////////////		
CX	//////////////////////////////////		
DX	//////////////////////////////////		

OUTPUTS: None

INT 10h, AH = BFh, AL = 02h - CHANGE CONTROL MODE

This interrupt changes the Control mode of the currently active video controller. This call is reserved for future use.

INPUTS:
AH = BFh
AL = 02h
BH = 05h = EGA (Enhanced Graphics Adapter)

AH	BFh	02h	AL
BH	05h	///////////////	BL
CX	//////////////////////////////////		
DX	//////////////////////////////////		

OUTPUTS: None

INT 10h, AH = BFh, AL = 03h - READ VIDEO ENVIRONMENT

This interrupt returns the Control mode, the active monitor selection, and the internal and external display types of the currently active video controller.

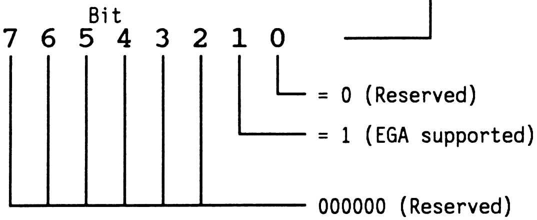
INPUTS:
AH = BFh
AL = 03h
BX = 0000h (See Note)

AH	BFh	03h	AL
BX	0000h		
CX	////////////////////////////////		
DX	////////////////////////////////		

Note: BX should be set to 0000h for full compatibility with previous video BIOS.

OUTPUTS:
BH = Active display
    00h = External
    01h = Internal
BL = Control Mode
    00h = Reserved
    05h = EGA
CH = 00h [Reserved]
CL = Modes Supported
DH = Internal Monitor Type
    00h = None
    01h = Dual-Mode Monochrome
DL = External Monitor Type
    00h = None
    01h = Dual-Mode Monochrome
    02h = RGBI 16-Color
    03h = Enhanced 64-Color

AH	/////////////////////////////////		AL
BH	ACT. DISPLAY	CONTROL MODE	BL
CH	////////////////	MODES SUPPORTED	CL
DH	INT. MON. TYPE	EXT. MON. TYPE	DL


Bit
7 6 5 4 3 2 1 0
= 0 (Reserved)
= 1 (EGA supported)
000000 (Reserved)

INT 10h, AH = BFh, AL = 04h - SET MODE-SWITCH DELAY

This interrupt enables or disables the 500-ms delay during mode-switch required for the COMPAQ Dual-Mode Monitor. The delay allows the monitor time to settle with video off after the mode change.

INPUTS:
AH = BFh
AL = 04h
BH = Switch
    00h = Enable delay
    01h = Disable delay

AH	BFh	04h	AL
BH	SWITCH	////////////////	BL
CX	/////////////////////////////////		
DX	/////////////////////////////////		

OUTPUTS: None

INT 1Fh - PTR - GRAPHICS DOT TABLE VECTOR

This vector points to the dot table used to generate and read graphics 8 x 8-dot characters in Modes 4, 5, and 6. This table is needed only for those characters within the range of 80h..FFh.

USE: It is used exclusively by the INT 10h video I/O routines, and then only in the graphics 8 x 8-dot modes for the upper-128 character set.

TABLE ENTRIES FOR 8 x 8-DOT CHARACTER CELLS

Alphanumeric characters are usually left-justified in the cell. Visible dots are usually composed of two adjacent pixels ON. Bottom row is normally blank (00h), except for descenders and special graphics characters. Figure 5-2 illustrates a sample character cell in the 8 x 8 character font.

8 x 8 Font

Scan Line	7	6	5	4	3	2	1	0	Contents
0	■	■	■	■	■	■	■		FEh
1	■	■				■	■		C6h
2	■				■	■			8Ch
3				■	■				18h
4			■	■			■		32h
5		■	■			■	■		66h
6	■	■	■	■	■	■	■		FEh
7									00h

Figure 5-2. Character Cell in 8 x 8 Character Font

Figure 5-3 illustrates the complete graphics table 8 x 8-dot table for the upper-128-character set.

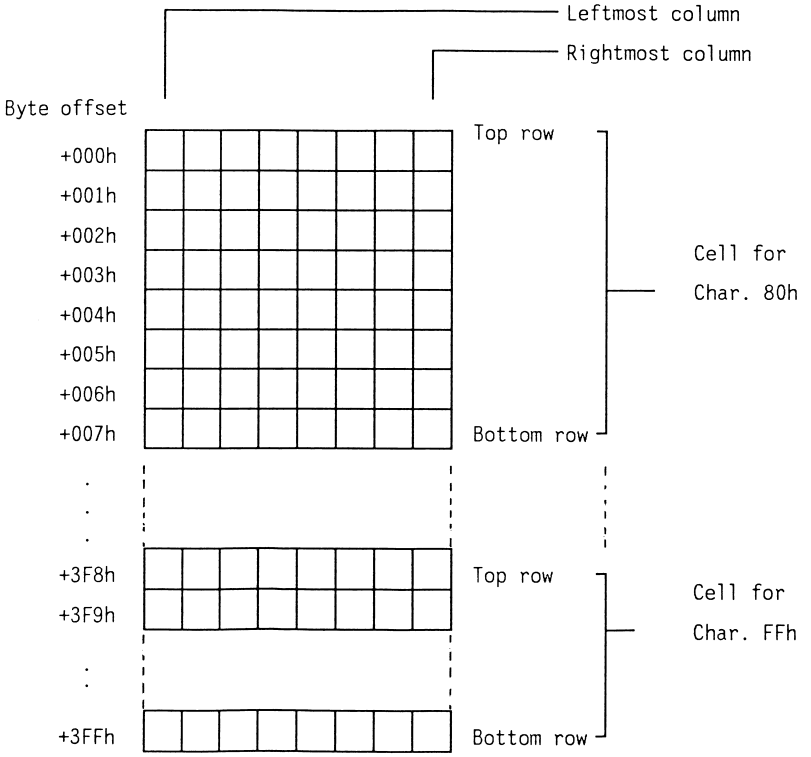

Leftmost column
Rightmost column
Byte offset
+000h
+001h
+002h
+003h
+004h
+005h
+006h
+007h
+3F8h
+3F9h
+3FFh
Top row
Bottom row
Cell for Char. 80h
Top row
Bottom row
Cell for Char. FFh

Figure 5-3. Layout of the Graphics 8 x 8-Dot Table for the Upper-128-Character Set

INT 42h - SW - COMPATIBLE VIDEO I/O

This interrupt gives direct access to previously available INT 10h functions. It saves the contents of the existing INT 10h vector during system power-on initialization and vectors function calls outside of the range 00h..13h and BFh to the system BIOS ROM. The alternate video adapter (if installed) is initialized into a video mode using this vector.

INITIALIZED: F000:F065

ROM ACTION: Transfers control to the system BIOS ROM video I/O entry point.

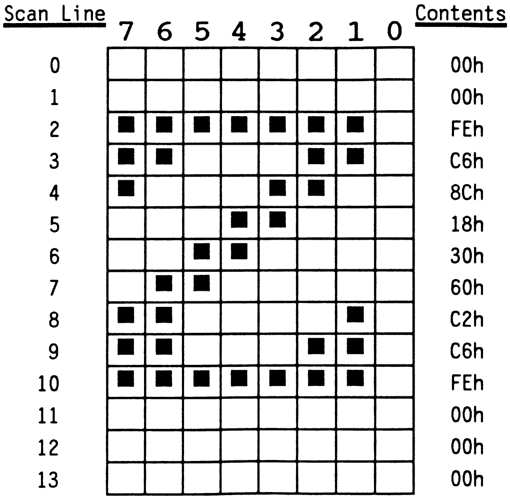
INT 43h - PTR - GRAPHICS DOT TABLE VECTOR

This vector points to the dot table used to generate and read dot graphics characters. In 8 x 14-dot modes, it points to the table for all characters. In the 8 x 8-dot modes, it points to the table for the first-128 characters (00h..7Fh).

Alphanumeric characters are usually left-justified in the cell. Visible dots are usually composed of two adjacent pixels ON. The top two (0, 1) and bottom three (11, 12, 13) scan lines are usually blank, except for ascenders, descenders, and special graphics characters.

Figure 5-4 illustrates a sample character cell in the 8 x 14 character font.

8 x 14 Font

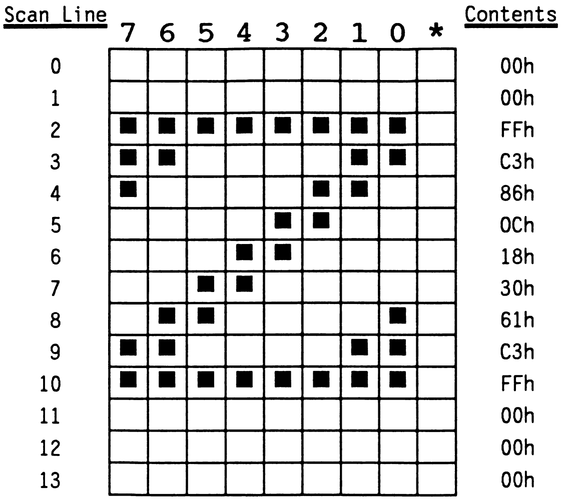


Scan Line	7	6	5	4	3	2	1	0	Contents
0									00h
1									00h
2	■	■	■	■	■	■	■		FEh
3	■	■				■	■		C6h
4	■				■	■			8Ch
5				■	■				18h
6			■	■					30h
7		■	■						60h
8	■	■					■		C2h
9	■	■				■	■		C6h
10	■	■	■	■	■	■	■		FEh
11									00h
12									00h
13									00h

Figure 5-4. Character Cell in 8 x 14 Character Font

Figure 5-5 illustrates a sample character cell in the 9 x 14 character font.

9 x 14 Font



Scan Line	7	6	5	4	3	2	1	0	*	Contents
0										00h
1										00h
2	■	■	■	■	■	■	■	■		FFh
3	■	■					■	■		C3h
4	■					■	■			86h
5					■	■				0Ch
6				■	■					18h
7			■	■						30h
8		■	■					■		61h
9	■	■					■	■		C3h
10	■	■	■	■	■	■	■	■		FFh
11										00h
12										00h
13										00h

*Pixels in the ninth column are supplied by the hardware.

Figure 5-5. Character Cell in the 9 x 14 Character Font

Figure 5-6 illustrates the complete graphics 8 x 14-dot table for the entire 256-character set.

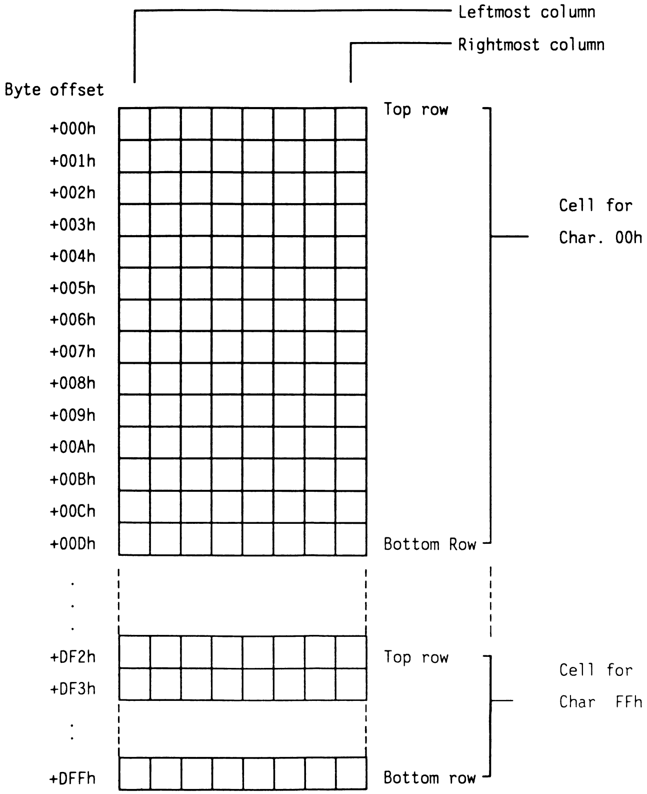

Leftmost column
Rightmost column
Byte offset
+000h
+001h
+002h
+003h
+004h
+005h
+006h
+007h
+008h
+009h
+00Ah
+00Bh
+00Ch
+00Dh
Top row
Bottom Row
Cell for
Char. 00h
+DF2h
+DF3h
+DFFh
Top row
Bottom row
Cell for
Char FFh

Figure 5-6. Layout of the Graphics 8 x 14-Dot Table for the Entire 256-Character Set

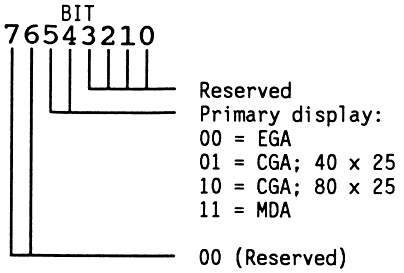
5.7 RAM AREA USED BY THE VIDEO BIOS

In addition to the RAM locations used by the interrupt vectors, the video BIOS maintains variables in RAM used for status information, pointers, and flags. The addresses of these locations are listed in Table 5-12.

Table 5-12. Video BIOS RAM Locations

Address	Name	Type	Description	Cross-Reference
0040:0010	equip	Word	System configuration	See Figure 5-7
0040:0049	v_mode	Byte	VDU Mode (00h..10h)	-
0040:004A	v_column	Word	Number of columns on screen	-
0040:004C	v_regen	Word	Length of screen regeneration buffer	-
0040:004E	v_start	Word	Starting address in regeneration buffer	-
0040:0050	v_cursav	Word array	Cursor area for each page (8 entries)	-
0040:0060	v_curmode	Word	Current cursor mode setting	-
0040:0062	v_page	Byte	Current displayed page	-
0040:0063	v_base	Word	Base address of video adapter board	-
0040:0065	v_modebit	Byte	Current mode	-
0040:0066	v_color	Byte	Current color	-
0040:0084	v_rows	Byte	Number of display rows - 1	-
0040:0085	v_points	Word	Bytes per character (1..32)	-
0040:0087	v_info	Byte	Miscellaneous information	See Figure 5-8
0040:0080	v_config	Byte	Configuration bits	See Figure 5-9
0040:00A8	v_save_ptr	Dword	Points to table of save areas	See Tables 5-13, 5-14, 5-15, 5-16, 5-17
0040:0100	pscrn_flag	Byte	Print screen status flag	See Figure 5-10

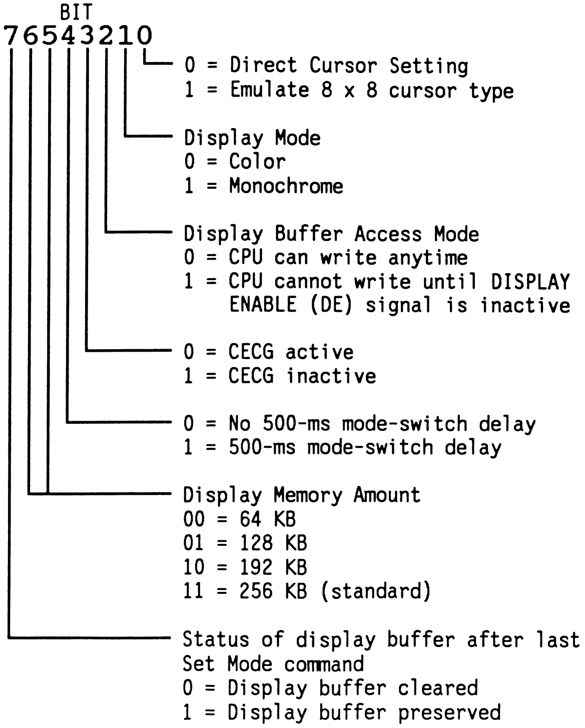
Figure 5-7 shows the the bit assignments and corresponding information for address 0040:0010 of Table 5-12.



Reserved
Primary display:
00 = EGA
01 = CGA; 40 x 25
10 = CGA; 80 x 25
11 = MDA
00 (Reserved)

Figure 5-7. Illustration of the Bit Assignments of the Word at Address 0040:0010

Figure 5-8 shows the bit assignment and corresponding information for address 0040:0087 of Table 5-12.



0 = Direct Cursor Setting
1 = Emulate 8 x 8 cursor type

Display Mode
0 = Color
1 = Monochrome

Display Buffer Access Mode
0 = CPU can write anytime
1 = CPU cannot write until DISPLAY ENABLE (DE) signal is inactive

0 = CECG active
1 = CECG inactive

0 = No 500-ms mode-switch delay
1 = 500-ms mode-switch delay

Display Memory Amount
00 = 64 KB
01 = 128 KB
10 = 192 KB
11 = 256 KB (standard)

Status of display buffer after last Set Mode command
0 = Display buffer cleared
1 = Display buffer preserved

Figure 5-8. Illustration of the Bit Assignments of the Byte at Address 0040:0087

Figure 5-9 shows the bit assignment and corresponding information for address 0040:0088 listed in Table 5-12.

```
   BIT
76543210
││││└┴┴┴─ Video System Configuration according
││││      to switch bank SW1 on the COMPAQ
││││      Enhanced Color Graphics Board:
││││
││││      Primary                        Secondary
││││      Adapter/Monitor                Adapter/Monitor
││││      0000 = MDA/Mono                EGA (40)/RGBI
││││      0001 = MDA/Mono                EGA (80)/RGBI
││││      0010 = MDA/Mono                EGA (x200)/RrGgBb
││││      0011 = MDA/Mono                EGA (x350)/RrGgBb
││││      0100 = CGA (40)/RGBI           EGA (720)/DM
││││      0101 = CGA (80)/RGBI           EGA (720)/DM
││││      0110 = CECG (40)/RGBI          MDA/Mono
││││      0111 = CECG (80)/RGBI          MDA/Mono
││││      1000 = CECG (x200)/RrGgBb      MDA/Mono
││││      1001 = CECG (x350)/RrGgBb      MDA/Mono
││││      1010 = CECG (720)/DM           CGA (40)/RGBI
││││      1011 = CECG (720)/DM           CGA (80)/RGBI
││││      1100 = Reserved                Reserved
││││      1101 = Reserved                Reserved
││││      1110 = Reserved                Reserved
││││      1111 = Reserved                Reserved
│││└───── Feature Bit 0 (FEAT0) w/Feature Output 1 (FC1) ON = TTL low
││└────── Feature Bit 1 (FEAT1) w/Feature Output 1 (FC1) ON = TTL low
│└─────── Feature Bit 0 (FEAT0) w/Feature Output 0 (FC0) ON = TTL low
└──────── Feature Bit 1 (FEAT1) w/Feature Output 0 (FC0) ON = TTL low
```

Legend: CECG = COMPAQ Enhanced Color Graphics Board
CGA = Color graphics adapter or COMPAQ Video Display Controller Board
DM = COMPAQ Dual-Mode Monitor
MDA = Monochrome display adapter or compatible
Mono = Compatible monochrome monitor
RGBI = 16-Color monitor
RrGgBb = COMPAQ Color Monitor or compatible enhanced color monitor

Figure 5-9. Illustration of the Bit Assignments of the Byte at Address 0040:0088

The parameter table pointers list provides a flexible means of substituting alternative operating parameters. This list, which is given in Table 5-13, consists of seven 32-bit (double-word or dword) pointers. These pointers in turn point to four tables of working EGA-mode parameters. The table column entitled "Parameter Reference" lists the sub-set of tables that list additional pertinent information.

Table 5-13. Listing of Parameter Table Pointers

Offset	Type	Pointer
+00h	Dword	Video Parameter Table (See Table 5-14)
+04h	Dword	Dynamic Save Area Table (See Table 5-15)
+08h	Dword	Text Mode Auxiliary Character Generator Table (See Table 5-16)
+0Ch	Dword	Graphics Mode Auxiliary Character Generator Table (See Table 5-17)
+10h	Dword	Reserved (=0000:0000)
+14h	Dword	Reserved (=0000:0000)
+18h	Dword	Reserved (=0000:0000)

Table 5-14 is a sub-set of Table 5-13 and represents the Video Parameter Table providing the offset and corresponding mode.

Table 5-14. Video Parameter Table
(1472 Bytes Total)

Offset	Mode
+000h	0
+040h	1
+080h	2
+0C0h	3
+100h	4
+140h	5
+180h	6
+1C0h	7
+200h	8
+240h	9
+280h	A
+2C0h	B
+300h	C
+340h	D
+380h	E
+3C0h	F (See Note)
+400h	10 (See Note)
+440h	F+
+480h	10+
+4C0h	0*
+500h	1*
+540h	2*
+580h	3*

Note: Because 256 KB of video RAM is standard on the COMPAQ Enhanced Color Graphics Board, Modes F+ (440h) and 10+ (480h) are always activated when Modes F and 10 are set, respectively. Therefore, offsets 3C0h and 400h are present, but not used.

Table 5-15 is a sub-set of Table 5-13 and represents the Dynamic Save Area Table providing the offset, type, and description of each entry.

Table 5-15. Dynamic Save Area Table

Offset	Type	Description
+00h	Byte	Number of columns
+01h	Byte	Number of rows - 1
+02h	Byte	Pixels per char. row
+03h	Word	Regeneration buffer size
+05h . . +08h	Byte array	Sequencer parameters (4 bytes) for Indices 01h..04h
+09h	Byte	Miscellaneous register
+0Ah . . +22h	Byte array	CRT controller parameters (25 bytes) for Indices 00h..18h
+23h . . +36h	Byte array	Attribute parameters (20 bytes) for Indices 00h..13h
+37h . . +3Fh	Byte array	Graphics parameters (9 bytes) for Indices 00h..08h

Table 5-16 is a sub-set of Table 5-13 and represents the Text Mode Auxiliary Character Generator Table providing the offset, type, and description of each entry.

Table 5-16. Text Mode Auxiliary Character Generator Table

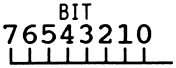
Offset	Type	Description
+00h	Byte	Bytes per character
+01h	Byte	Block to load
+02h	Word	Count to store (0 = Full character set = 256)
+04h	Word	Character offset
+06h	Dword	Font table pointer
+0Ah	Byte	Number of displayed rows (0 = Maximum capable)
+0Bh	Byte	1st mode value
+0Ch	Byte	2nd mode value
:	:	:
	Byte	Mode value list terminator = FFh

Table 5-17 is a sub-set of Table 5-13 and represents the Graphics Mode Auxiliary Character Generator Table providing the offset, type, and description of each entry.

Table 5-17. Graphics Mode Auxiliary Character Generator Table

Offset	Type	Description
+00h	Byte	Number of displayable rows (0 = Maximum capable)
+01h	Word	Bytes per character
+03h	Dword	Font table pointer
+07h	Byte	1st mode value
+08h	Byte	2nd mode value
:	:	:
	Byte	Mode value list Terminator = FFh

Figure 5-10 shows the bit assignment and corresponding information for address 0040:0100 of Table 5-12.



Valid flag values:
00000000 = Idle
00000001 = Printing
11111111 = Error (printing locked out)

Figure 5-10. Illustration of the Bit Assignment of the Byte at Address 0040:0100

TABLE OF CONTENTS

CHAPTER 6
CONFIGURING THE BOARD

6.1 VIDEO MEMORY SPACING AND I/O ADDRESSING 6-1

6.2 SWITCHES AND JUMPERS 6-2

- Switch Bank SW1 6-3
- Switch Bank SW2 6-6
- Jumper P1 6-7
- Jumper JP3 6-7

6.3 VIDEO CONFIGURATIONS FOR PORTABLE COMPUTERS 6-8

6.4 VIDEO CONFIGURATIONS FOR DESKTOP COMPUTERS 6-9

6.5 CONNECTORS 6-10

- J1 and J2, Dual RCA Connectors 6-10
- J4, Special Features 6-11
- J401, Internal Monitor 6-13
- J404, External Monitor 6-14
- J405, Lightpen Interface 6-16

6.6 OUTPUT SIGNAL WAVEFORMS 6-17

Chapter 6
CONFIGURING THE BOARD

This chapter provides configuration information for the COMPAQ Enhanced Color Graphics Board as the only video display controller board in a system and as the primary video controller board in a dual-controller setup. The configuration is also given for the board as the secondary controller in a dual-controller setup. The chapter is organized in the following manner:

- Explanation of video memory spacing and I/O addressing
- Switch and jumper descriptions for the board as the only/primary video controller
- Switch and jumper descriptions for the board as the secondary video controller
- Installation guide for the board in COMPAQ Portable Computers
- Installation guide for the board in COMPAQ Desktop Computers
- Connectors and corresponding pin-signal descriptions
- Output signal waveforms

6.1 VIDEO MEMORY SPACING AND I/O ADDRESSING

Compatibility requirements dictate that certain guidelines must be adhered to when installing and operating video display controller boards in a personal computer. Standard memory and I/O addresses have been established for each functional classification of controller, and only one controller can occupy a memory or I/O space at a time. Therefore, it is impossible to operate multiple video display controllers of the same classification in one system.

For example, two CGA-class controllers cannot simultaneously exist in a personal computer. This duality would result in conflicting memory and I/O addressing assignments. Thus, it is important to recognize standard memory and I/O addressing assignments for the various classes of video controllers.

- CGA-class controllers, such as the COMPAQ Video Display Controller Board, occupy 16 KB of memory space from B8000h to BBFFFh and I/O addresses 3D4h to 3DCh.
- MDA-class controllers occupy 16 KB of memory space from B0000h to B3FFFh and I/O addresses 3B4h to 3BCh.
- EGA-class controllers, such as the COMPAQ Enhanced Color Graphics Board, are unique in that they operate in four different ranges, depending on the emulation for which they are configured. The ranges are described in Table 6-1.

Table 6-1. Operating Address Ranges for EGA-Class Controllers According to Emulation

Emulation	Amount of Memory (In KB)	Memory Space	I/O Address
Color Graphics	32	B8000h to BFFFFh	3C0h to 3DCh
Monochrome	32	B0000h to B7FFFh	3B4h to 3CAh
Enhanced Color Graphics	64	A0000h to AFFFFh	3C0h to 3DCh
Enhanced Monochrome Graphics	64	A0000h to AFFFFh	3B4h to 3CAh

A unique feature of the COMPAQ Enhanced Color Graphics Board, when used in conjunction with the COMPAQ MS DOS Version 3 MODE command, is that the board can be switched between color and monochrome emulations. Configuration switch bank SW1 only determines the mode in which the system operates at power-on.

6.2 SWITCHES AND JUMPERS

The COMPAQ Enhanced Color Graphics Board has two switch banks (SW1 and SW2) and two jumpers (P1 and JP3) to allow configuration flexibility across COMPAQ product lines and to obtain the highest level of compatibility. The switch bank and jumper locations are shown in Figure 6-1.

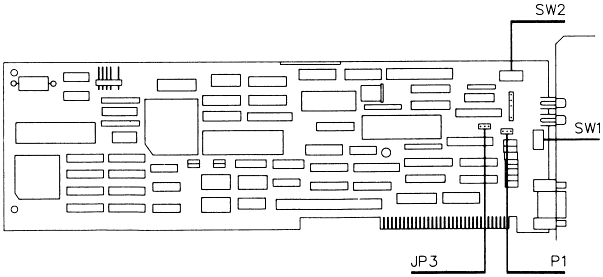


Figure 6-1. Switch Bank and Jumper Locations

Switch Bank SW1

Switch bank SW1 is the only external switch accessible on the I/O access cover below a dual RCA connector (J1,J2). SW1 settings determine configuration of the board in three respects: configuration as primary/secondary video display controller, text resolution, and character format at power-on. Positions 1,2,3, and 4 must be encoded to establish the configuration.

Table 6-2 gives the settings for SW1 when the COMPAQ Enhanced Color Graphics Board is the only video display controller installed in the system.

Table 6-2. SW1 Settings with Only the COMPAQ Enhanced Color Graphics Board in the System

SW1 Setting 1	2	3	4	Mode	Initial Text Resolution	Char. Format	Monitor Type
ON	OFF	OFF	ON	Color	320 x 200	40 x 25	COMPAQ Dual-Mode; or Compatible RGBI Color
OFF	OFF	OFF	ON	Color	640 x 200	80 x 25	COMPAQ Dual-Mode; or Compatible RGBI Color
ON	ON	ON	OFF	Color	640 x 200	80 x 25	COMPAQ Color or Dual-Mode
OFF	ON	ON	OFF	Color	640 x 350	80 x 25	COMPAQ Color or Dual-Mode
ON	OFF	ON	OFF	Mono.	720 x 350	80 x 25	COMPAQ Dual-Mode
OFF	OFF	ON	OFF	Mono.	720 x 350	80 x 25	COMPAQ Dual-Mode

Table 6-3 gives the settings for configuration switch bank SW1 when the COMPAQ Enhanced Color Graphics Board is the primary video display controller in a system with two video display controller boards.

Table 6-3. SW1 Settings for Two Video Display Controllers in the Same System with the COMPAQ Enhanced Color Graphics Board as the Primary Controller

	PRIMARY CONTROLLER COMPAQ Enhanced Color Graphics Board				SECONDARY CONTROLLER			
SW1 Setting 1 2 3 4	Mode	Initial Text Resolution	Char. Format	Monitor Type	Video Controller Type	Initial Text Resolution	Char. Format	Monitor Type
ON OFF OFF ON	Color	320 x 200	40 x 25	COMPAQ Dual-Mode; or RGBI Color	MDA	720 x 350	80 x 25	Monochrome
OFF OFF OFF ON	Color	640 x 200	80 x 25	COMPAQ Dual-Mode; or RGBI Color	MDA	720 x 350	80 x 25	Monochrome
ON ON ON OFF	Color	640 x 200	80 x 25	COMPAQ Color or Dual-Mode	MDA	720 x 350	80 x 25	Monochrome
OFF ON ON OFF	Color	640 x 350	80 x 25	COMPAQ Color or Dual-Mode	MDA	720 x 350	80 x 25	Monochrome
ON OFF ON OFF	Mono.	720 x 350	80 x 25	COMPAQ Dual-Mode	COMPAQ VDU or a CGA	320 x 200	40 x 25	RGBI Color
OFF OFF ON OFF	Mono.	720 x 350	80 x 25	COMPAQ Dual-Mode	COMPAQ VDU or a CGA	640 x 200	80 x 25	RGBI Color

Legend: MDA = Monochrome Display Adapter
VDU = Video Display Controller
CGA = Color Graphics Adapter
RGBI = 16-Color Monitor

Table 6-4 gives the settings for configuration switch bank SW1 when the COMPAQ Enhanced Color Graphics Board is the secondary video display controller in a system with two video display controller boards.

Table 6-4. SW1 Settings for Two Video Display Controllers in the Same System with the COMPAQ Enhanced Color Graphics Board as the Secondary Controller

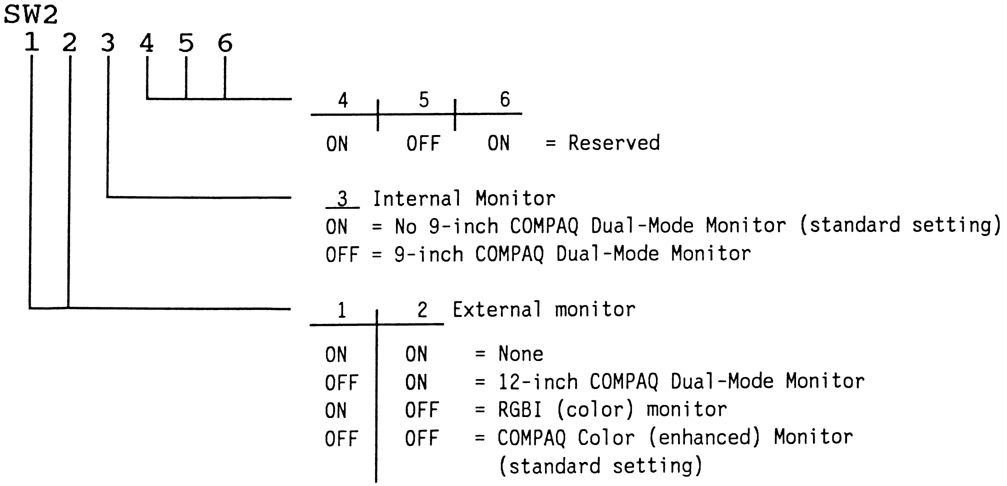
	SECONDARY CONTROLLER COMPAQ Enhanced Color Graphics Board				PRIMARY CONTROLLER			
SW1 Setting 1 2 3 4	Mode	Initial Text Resolution	Char. Format	Monitor Type	Video Controller Type	Initial Text Resolution	Char. Format	Monitor Type
ON ON ON ON	Color	320 x 200	40 x 25	COMPAQ Dual-Mode; or RGBI Color	MDA	720 x 350	80 x 25	Monochrome
OFF ON ON ON	Color	640 x 200	80 x 25	COMPAQ Dual-Mode; or RGBI Color	MDA	720 x 350	80 x 25	Monochrome
ON OFF ON ON	Color	640 x 200	80 x 25	COMPAQ Color or Dual-Mode	MDA	720 x 350	80 x 25	Monochrome
OFF OFF ON ON	Color	640 x 350	80 x 25	COMPAQ Color or Dual-Mode	MDA	720 x 350	80 x 25	Monochrome
ON ON OFF ON	Mono.	720 x 350	80 x 25	COMPAQ Dual-Mode	COMPAQ VDU or a CGA	320 x 200	40 x 25	RGBI Color
OFF ON OFF ON	Mono.	720 x 350	80 x 25	COMPAQ Dual-Mode	COMPAQ VDU or a CGA	640 x 200	80 x 25	RGBI Color

Legend:
MDA = Monochrome Display Adapter
VDU = Video Display Controller
CGA = Color Graphics Adapter
RGBI = 16-Color Monitor

Switch Bank SW2

The setting of configuration switch bank SW2 is determined by the type of monitor(s) attached to the external and/or internal connector. Positions 1 and 2 reference the type of external monitor attached, while position 3 references whether the internal monitor connector is being used. Positions 4,5, and 6 are reserved for future usage.

Figure 6-2 illustrates the setting for SW2.



4	5	6	
ON	OFF	ON	= Reserved

3 Internal Monitor
ON = No 9-inch COMPAQ Dual-Mode Monitor (standard setting)
OFF = 9-inch COMPAQ Dual-Mode Monitor

1	2	External monitor
ON	ON	= None
OFF	ON	= 12-inch COMPAQ Dual-Mode Monitor
ON	OFF	= RGBI (color) monitor
OFF	OFF	= COMPAQ Color (enhanced) Monitor (standard setting)

Figure 6-2. SW2 Configuration Settings

Jumper P1

Jumper P1 provides the ability to select between secondary red (r) for enhanced color monitors and ground (Gnd) for the COMPAQ Dual-Mode Monitor and RGB and RGBI monitors on pin 2 of connector J404, the external monitor connector.

Table 6-5 illustrates jumper P1 settings.

Table 6-5. Jumper P1 Settings

Jumper	Connection	Function
P1	1 - 2	Secondary red (r) is provided on pin 2 (standard setting). Used for COMPAQ Color Monitor or compatible.
	2 - 3	Ground (Gnd) is provided on pin 2. Used for COMPAQ Dual-Mode Monitor, an RGBI monitor, or an RGBI monitor.

Jumper JP3

Jumper JP3 is provided for the base address to alternate between 3xxh and 2xxh addresses for special applications in which two COMPAQ Enhanced Color Graphics Boards are installed in one system.

Table 6-6 illustrates jumper JP3 settings.

Table 6-6. Jumper JP3 Settings

Jumper	Connection	Function
JP3	1 - 2	3xxh address select (standard setting)
	2 - 3	2xxh address select

Note: Video BIOS does not support address 2xxh.

6.3 VIDEO CONFIGURATIONS FOR PORTABLE COMPUTERS

Table 6-7 lists all possible video configurations for a COMPAQ Portable Computer.

Table 6-7. SW1 and SW2 Settings for All Possible Video Configurations for COMPAQ Portable Computers

Config. No.	Primary (or Only) Video Controller/ Mode	Initial Text Resolution	Char. Format	Internal Monitor	External Monitor	(Optional) Secondary Video Controller/ Mode	Initial Text Resolution	Char. Format	Monitor	SW1 Setting				SW2 Setting						J1
										1	2	3	4	1	2	3	4	5	6	
1	CECG Board/ Enhanced Color	640 x 350	80 x 25	DM	COMPAQ Color	MDA	720 x 350	80 x 25	Compat. Mono-chrome	OFF	ON	ON	OFF	OFF	OFF	OFF	ON	OFF	ON	1-2
2					DM (See Note)					OFF	ON	ON	OFF	OFF	ON	OFF	ON	OFF	ON	2-3
3					None					OFF	ON	ON	OFF	ON	ON	OFF	ON	OFF	ON	2-3
4		640 x 200	80 x 25	DM	COMPAQ Color	MDA	720 x 350	80 x 25	Compat. Mono-chrome	ON	ON	ON	OFF	OFF	OFF	OFF	ON	OFF	ON	1-2
5					DM (See Note)					ON	ON	ON	OFF	OFF	ON	OFF	ON	OFF	ON	2-3
6					None					ON	ON	ON	OFF	ON	ON	OFF	ON	OFF	ON	2-3
7	CECG Board/ Mono-chrome	720 x 350	80 x 25	DM	DM (See Note)	VDU or CGA	640 x 200	80 x 25	RGBI Color	OFF	OFF	ON	OFF	OFF	ON	OFF	ON	OFF	ON	2-3
8					None					OFF	OFF	ON	OFF	ON	ON	OFF	ON	OFF	ON	2-3
9					DM (See Note)		320 x 200	40 x 25	RGBI Color	ON	OFF	ON	OFF	OFF	ON	OFF	ON	OFF	ON	2-3
10					None					ON	OFF	ON	OFF	ON	ON	OFF	ON	OFF	ON	2-3
11	CECG Board/ Enhanced Color	640 x 200	80 x 25	DM	RGBI Color	MDA	720 x 350	80 x 25	Compat. Mono-chrome	OFF	OFF	OFF	ON	ON	OFF	OFF	ON	OFF	ON	2-3
12					DM (See Note)					OFF	OFF	OFF	ON	OFF	ON	OFF	ON	OFF	ON	2-3
13					None					OFF	OFF	OFF	ON	ON	ON	OFF	ON	OFF	ON	2-3
14		320 x 200	40 x 25	DM	RGBI Color	MDA	720 x 350	80 x 25	Compat. Mono-chrome	ON	OFF	OFF	ON	ON	OFF	OFF	ON	OFF	ON	2-3
15					DM (See Note)					ON	OFF	OFF	ON	OFF	ON	OFF	ON	OFF	ON	2-3
16					None					ON	OFF	OFF	ON	ON	ON	OFF	ON	OFF	ON	2-3

Legend:
CECG = COMPAQ Enhanced Color Graphics
DM = COMPAQ Dual-Mode Monitor
MDA = Monochrome Display Adapter
VDU = COMPAQ Video Display Controller
CGA = Color Graphics Adapter
RGBI = 16-Color Monitor

Note: The COMPAQ Dual-Mode Monitor cannot be connected to COMPAQ Portable Computers as an external monitor.

6.4 VIDEO CONFIGURATIONS FOR DESKTOP COMPUTERS

Table 6-8 lists all possible video configurations for a COMPAQ Desktop Computer.

Table 6-8. SW1 and SW2 Settings for All Possible Video Configurations for COMPAQ Desktop Computers

Config. No.	Primary (or Only) Video Controller/ Mode	Initial Text Resolution	Char. Format	Monitor	(Optional) Secondary Video Controller/ Mode	Initial Text Resolution	Char. Format	Monitor	SW1 Setting				SW2 Setting						J1
									1	2	3	4	1	2	3	4	5	6	
1	CECG Board/ Enhanced Color	640 x 350	80 x 25	COMPAQ Color	MDA	720 x 350	80 x 25	Compat. Mono-chrome	OFF	ON	ON	OFF	OFF	OFF	ON	ON	OFF	ON	1-2
2		640 x 200							ON	ON	ON	OFF	OFF	OFF	ON	ON	OFF	ON	1-2
3		640 x 350		DM					OFF	ON	ON	OFF	OFF	ON	ON	ON	OFF	ON	2-3
4		640 x 200							ON	ON	ON	OFF	OFF	ON	ON	ON	OFF	ON	2-3
5	CECG Board/ Mono.	720 x 350	80 x 25	DM	VDU or CGA	640 x 200	80 x 25	RGBI Color	OFF	OFF	ON	OFF	OFF	ON	ON	ON	OFF	ON	2-3
6				VDU or CGA		320 x 200	40 x 25		ON	OFF	ON	OFF	OFF	ON	ON	ON	OFF	ON	2-3
7	CECG Board/ Enhanced Color	640 x 200	80 x 25	DM	MDA	720 x 350	80 x 25	Compat. Mono-chrome	OFF	OFF	OFF	ON	OFF	ON	ON	ON	OFF	ON	2-3
8		320 x 200							ON	OFF	OFF	ON	OFF	ON	ON	ON	OFF	ON	2-3
9		640 x 200		RGBI Color					OFF	OFF	OFF	ON	ON	OFF	ON	ON	OFF	ON	2-3
10		320 x 200		MDA					ON	OFF	OFF	ON	ON	OFF	ON	ON	OFF	ON	2-3
11	MDA	720 x 350	80 x 25	Compat. Mono-chrome	CECG Board/ Enhanced Color	640 x 350	80 x 25	COMPAQ Color	OFF	OFF	ON	ON	OFF	OFF	ON	ON	OFF	ON	1-2
12						640 x 200			ON	OFF	ON	ON	OFF	ON	ON	ON	OFF	ON	1-2
13						640 x 350		DM	OFF	OFF	ON	ON	OFF	ON	ON	ON	OFF	ON	2-3
14						640 x 200			ON	OFF	ON	ON	OFF	ON	ON	ON	OFF	ON	2-3
15	VDU or CGA	640 x 200	80 x 25	RGBI Color	CECG Board/ Mono.	720 x 350	80 x 25	DM	OFF	ON	OFF	ON	OFF	ON	ON	ON	OFF	ON	2-3
16		320 x 200	40 x 25						ON	ON	OFF	ON	OFF	ON	ON	ON	OFF	ON	2-3
17	MDA	720 x 350	80 x 25	Compat. Mono-chrome	CECG Board/ Enhanced Color	640 x 200	80 x 25	DM	OFF	ON	ON	ON	OFF	ON	ON	ON	OFF	ON	2-3
18						320 x 200	40 x 25		ON	ON	ON	ON	OFF	ON	ON	ON	OFF	ON	2-3
19						640 x 200	80 x 25	RGBI Color	OFF	ON	ON	ON	ON	OFF	ON	ON	OFF	ON	2-3
20						320 x 200	40 x 25		ON	ON	ON	ON	ON	OFF	ON	ON	OFF	ON	2-3

Legend: CECG = COMPAQ Enhanced Color Graphics
DM = COMPAQ Dual-Mode Monitor
MDA = Monochrome Display Adapter
VDU = COMPAQ Video Display Controller
CGA = Color Graphics Adapter
RGBI = 16-Color Monitor

6.5 CONNECTORS

Tables 6-9 through 6-15 list the signal descriptions for the connectors on the COMPAQ Enhanced Color Graphics Board. Figure 6-3 shows the locations of the connectors. Figures 6-4 through 6-10 illustrate the connectors.

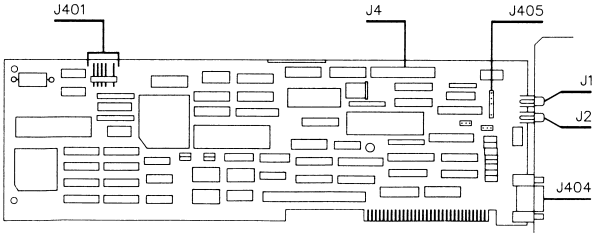

J401
J4
J405
J1
J2
J404

Figure 6-3. Connector Locations

J1 and J2, Dual RCA Connectors

Connectors J1 and J2 are connectors that are used in conjunction with a Special Features board, if one is installed in connector J4. J1 and J2 have no function unless a Special Features board is driving signals through them. Table 6-9 lists the signals for connectors J1 and J2.

Table 6-9. J1 and J2 Signals

Signal	Pin	I/O	Description
J1-1 and J2-1	1	I/O	This signal is determined by the board connected to the Special Features board connector J4.
Chassis Ground	2	-	Chassis Ground

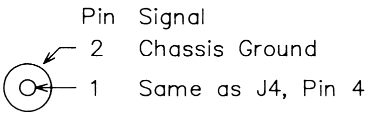

Pin
Signal
2
Chassis Ground
1
Same as J4, Pin 4

Figure 6-4. J1 RCA Connector

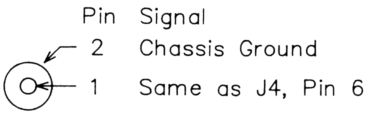

Pin
Signal
2
Chassis Ground
1
Same as J4, Pin 6

Figure 6-5. J2 RCA Connector

J4, Special Features

The Special Features connector is provided to support the development of additional video functions, such as composite video output and future circuits.

Note: The I/O descriptions given for signals in Table 6-10 are relative to a board plugged into Special Features connector J4. Therefore, signals driven by the COMPAQ Enhanced Color Graphics Board are designated as inputs (I).

Table 6-10. J4 Connector Signals

Signal	Pin	I/O	Description
+12 VDC	3	I	Power inputs for Special Features board
+5 VDC	32	I	
-12 VDC	2	I	
14MHZ-	26	I	14.31818-MHz clock
ATRS/L	9	I	Character clock. High during shifting, low during loading
BLANK	18	I	Active during vertical and horizontal sync
Signals from video PLD:			
BUFB	14	I	Primary blue (B)
BUFBS/9D	23	I	Secondary blue (b) or MODE for the COMPAQ Dual-Mode Monitor
BUFG	12	I	Primary green (G)
BUFGS/I	22	I	Secondary green (g) or intensity (I)
BUFHS	24	I	Horizontal sync
BUFR	15	I	Primary red (R)
BUFRS	13	I	Secondary red (r)
BUFVS	25	I	Vertical sync
Signals to J404:			
EXTB	10	0	Primary blue (B), pin 5
EXTBS/9D	8	0	Secondary blue (b) or MODE for the COMPAQ Dual-Mode Monitor, pin 7
EXTG	11	0	Primary green (G), pin 4
EXTGS/I	6	0	Secondary green (g) or intensity (I), pin 6
EXTHS	30	0	Horizontal sync, pin 8
EXTR	16	0	Primary red (R), pin 3
EXTRS	7	0	Secondary red (r), pin 2
EXTVS	29	0	Vertical sync, pin 9
EXTOSC	28	0	An alternate source for the master clock

(Continued)

Table 6-10. J4 Connector Signals (Concluded)

Signal	Pin	I/O	Description
Output from the Feature Control register (3C2h):			
FC0	21	I	Bit <0>
FC1	20	I	Bit <1>
Signals to the Input Status 0 register (3xAh):			
FEAT0	19	0	Bit <5>
FEAT1	17	0	Bit <6>
FEATEN	27	I	Enables the outputs of the Special Features board
GND	1,31	-	Ground
External video I/O from/to RCA connectors:			
J1-1	4	I/O	J1
J2-1	5	I/O	J2

Figure 6-6 shows the Special Features connector.

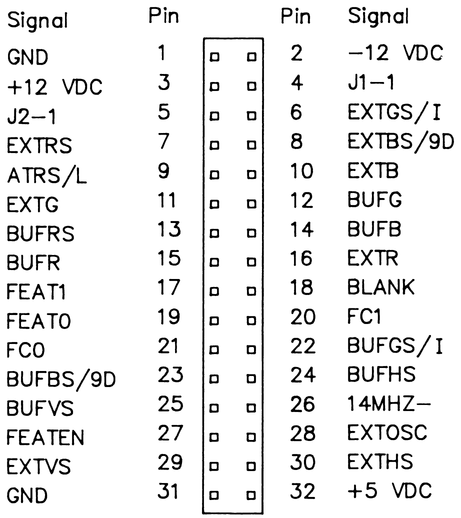

Signal
Pin
Pin
Signal
GND 1 2 −12 VDC
+12 VDC 3 4 J1−1
J2−1 5 6 EXTGS/I
EXTRS 7 8 EXTBS/9D
ATRS/L 9 10 EXTB
EXTG 11 12 BUFG
BUFRS 13 14 BUFB
BUFR 15 16 EXTR
FEAT1 17 18 BLANK
FEAT0 19 20 FC1
FC0 21 22 BUFGS/I
BUFBS/9D 23 24 BUFHS
BUFVS 25 26 14MHZ−
FEATEN 27 28 EXTOSC
EXTVS 29 30 EXTHS
GND 31 32 +5 VDC

Figure 6-6. J4, Special Features Connector

J401, Internal Monitor

Table 6-11 describes the signals by pin number for connector J401.

Table 6-11. J401 Connector Signals

Signal	Pin	I/O	Description
Chassis Ground	11,12	-	Cable shield
HS	7	0	Horizontal sync is an active-high TTL signal. (For timing waveforms, refer to Section 6.4.)
MODE	3	0	This TTL signal controls the monitor mode. During a mode change, both HS and VS- are suppressed for as many as 16 vertical frames. This signal is high (=1) for the low-scan mode and low (=0) for the high-scan mode.
Signal Ground	2,4,8	-	This is the ground reference for the TTL signals.
VID	5	0	This analog video output signal ranges from 1.0 VDC (black) to 2.5 VDC (white) with an impedance range of from 470 ohms to 6800 ohms. Its ground reference is the Video Ground signal (pin 6).
Video Ground	6	-	This is the ground reference for the VID signal (pin 5).
VS-	1	0	Vertical sync is an active-low TTL signal. (For timing waveforms, refer to Section 6.4.)

Figure 6-7 shows the internal monitor connector.

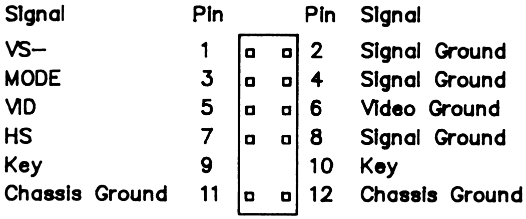

Signal
Pin
Pin
Signal
VS–
1
2
Signal Ground
MODE
3
4
Signal Ground
VID
5
6
Video Ground
HS
7
8
Signal Ground
Key
9
10
Key
Chassis Ground
11
12
Chassis Ground

Figure 6-7. J401, Internal Monitor Connector

J404, External Monitor

The signal description for the external monitor depends upon the type of external monitor connected. Table 6-12 compares the signal functions for the various types of monitors supported.

Table 6-12. Comparison of Signal Functions

Pin	COMPAQ Dual-Mode or a Compatible RGBI Monitor	COMPAQ Color or a Compatible Enhanced Color Monitor
1	Ground	Ground
2	Ground	Secondary red (r)
3	Red	Primary red (R)
4	Green	Primary green (G)
5	Blue	Primary blue (B)
6	Intensity	Secondary green (g) or Intensity (I)
7	Scan Mode	Secondary blue (b)
8	Horizontal sync	Horizontal sync
9	Vertical sync	Vertical sync
Shell	Chassis ground	Chassis ground

Tables 6-13 and 6-14 give the signal descriptions for external monitor connectors for the COMPAQ Dual-Mode Monitor and compatible color monitors, respectively. All signals are active-high TTL signals unless otherwise noted.

Table 6-13. J404 Connector Signals for the COMPAQ Dual-Mode Monitor or a Compatible RGBI Color Monitor

Signal	Pin	I/O	Description
B	5	0	Blue
G	4	0	Green
HS	8	0	Horizontal sync (Refer to Figure 6-12 for timing characteristics.)
I	6	0	Intensity for COMPAQ Dual-Mode or RGBI Monitor.
MODE	7	0	This signal specifies the monitor high- or low-scan mode. The MODE signal is high for the low-scan mode and low for the high-scan mode.
R	3	0	Red
Signal Ground	1,2	-	TTL signal reference
VS	9	0	Vertical sync (Refer to Figure 6-13 for timing characteristics.)

Figure 6-8 shows the external monitor connector for the COMPAQ Dual-Mode Monitor or a compatible RGBI Color Monitor.

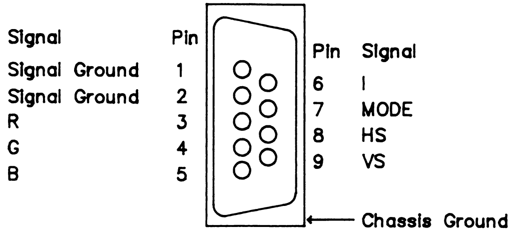

Signal
Pin
Signal Ground 1
Signal Ground 2
R 3
G 4
B 5
Pin Signal
6 I
7 MODE
8 HS
9 VS
Chassis Ground

Figure 6-8. J404 Connector for the COMPAQ Dual-Mode Monitor or a Compatible RGBI Color Monitor

Table 6-14. J404 Connector Signals for the COMPAQ Color Monitor or a Compatible Enhanced Color Monitor

Signal	Pin	I/O	Description
B	5	0	Primary blue (B)
G	4	0	Primary green (G)
HS	8	0	Horizontal sync (Refer to Figure 6-12 for timing characteristics.)
g/I	6	0	Secondary green (g) for COMPAQ Color or RrGgBb Monitor
b	7	0	Secondary blue (b)
r	2	0	Secondary red (r)
R	3	0	Primary red (R)
Signal Ground	1	-	TTL signal reference
VS	9	0	Vertical sync is active-high for the 200-scan line mode and active-low for the 350-scan line mode. (Refer to Figure 6-13 for timing characteristics.)

Figure 6-9 shows the external monitor connector.

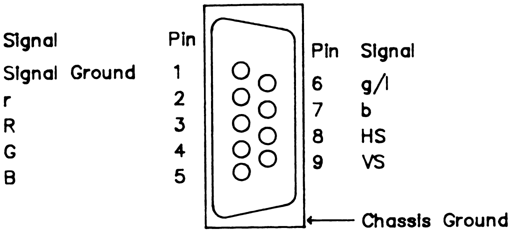

Signal
Pin
Signal Ground 1
r 2
R 3
G 4
B 5
Pin
Signal
6 g/I
7 b
8 HS
9 VS
Chassis Ground

Figure 6-9. J404 Connector for the COMPAQ Color Monitor or a Compatible Enhanced Color Monitor

J405, Lightpen Interface

Table 6-15 gives the signal descriptions for the lightpen interface.

Table 6-15. J405, Lightpen Interface Signals

Signal	Pin	I/O	Description
+5 VDC	5	0	+5 VDC lightpen power source
+12 VDC	6	0	+12 VDC lightpen power source
LIGHTPEN SWITCH-	3	I	Active-low signal indicates lightpen switch contact closed
LIGHTPEN TRIGGER-	1	I	Active-low signal indicates lightpen triggered
Signal Ground	4	-	Reference for TTL signals

Figure 6-10 shows the lightpen interface connector.

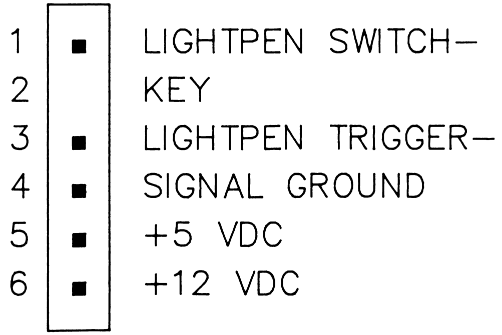
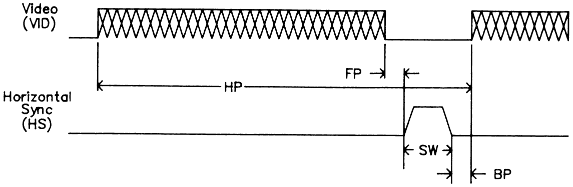

1 LIGHTPEN SWITCH-
2 KEY
3 LIGHTPEN TRIGGER-
4 SIGNAL GROUND
5 +5 VDC
6 +12 VDC

Figure 6-10. J405, Lightpen Interface Connector

6.6 OUTPUT SIGNAL WAVEFORMS

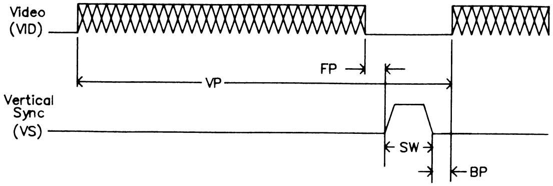
Figure 6-11 shows the timing characteristics for the horizontal sync (HS) signals.



		COMPAQ Dual-Mode		COMPAQ Color	
Symbol	Parameter	High Scan	Low Scan	350 Scan Line	200 Scan Line
HP	Horizontal Period	54.1 µs	63.7 µs	45.8µs	63.7 µs
FP	Front Porch	5.7 µs	6.7 µs	0 µs	6.7 µs
SW	Sync Width	3.8 µs	4.5 µs	4.9µs	4.5 µs
BP	Back Porch	6.6 µs	7.8 µs	1.5µs	7.8 µs

Figure 6-11. Horizontal Timing

Figure 6-12 shows the timing characteristics for the vertical sync (VS) signals.



		COMPAQ Dual-Mode		COMPAQ Color	
Symbol	Parameter	High Scan	Low Scan	350 Scan Line	200 Scan Line
VP	Vertical Period	20.0 ms	16.7 ms	16.7 ms	16.7 ms
FP	Front Porch	0 ms	1.5 ms	0 ms	1.5 ms
SW	Sync Width	0.2 ms	0.2 ms	0.6 ms	0.2 ms
BP	Back Porch	0.9 ms	2.2 ms	0.1 ms	2.2 ms

Figure 6-12. Vertical Timing

Figure 6-13 shows the D/A conversion characteristics for the COMPAQ Dual-Mode Monitor video display. (Refer to Chapter 2, Tables 2-1 and 2-2, for more specific information about relative intensities of this monitor.)

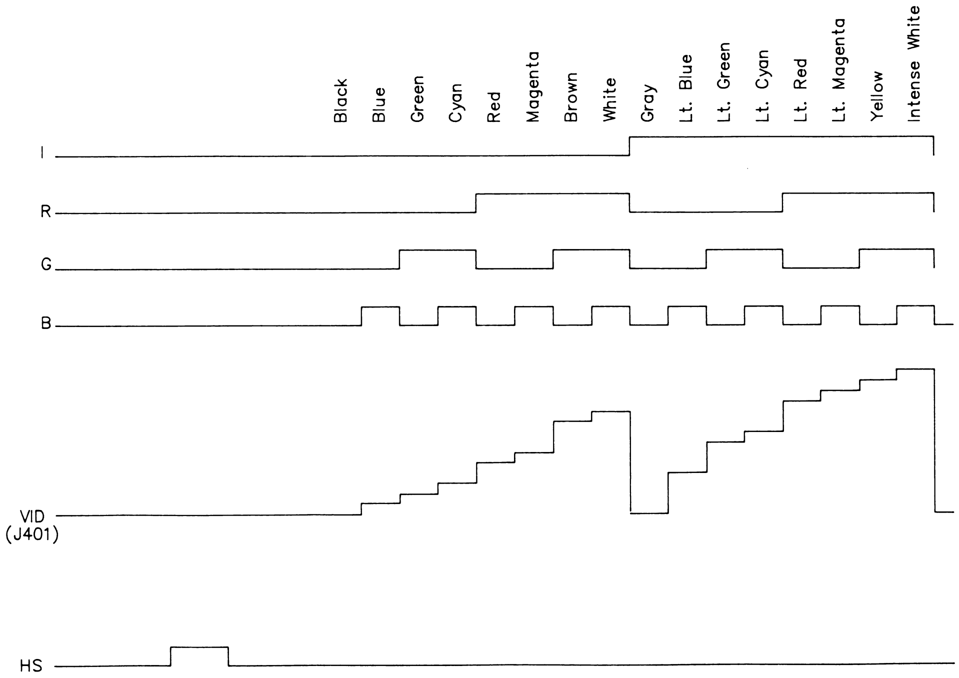

Black
Blue
Green
Cyan
Red
Magenta
Brown
White
Gray
Lt. Blue
Lt. Green
Lt. Cyan
Lt. Red
Lt. Magenta
Yellow
Intense White
I
R
G
B
VID
(J401)
HS

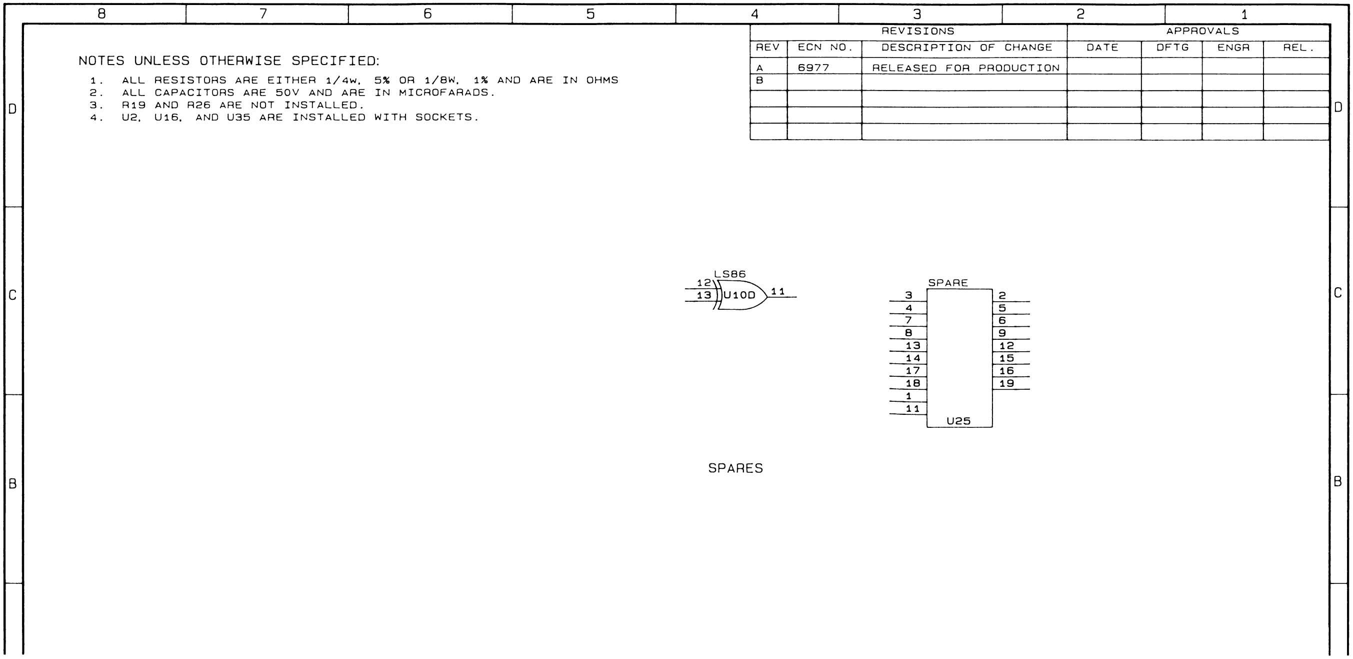
Figure 6-13. Digital-to-Analog Conversion Characteristics for the COMPAQ Dual-Mode Monitor Video Display

TABLE OF CONTENTS

CHAPTER 7
SCHEMATICS

Chapter 7
SCHEMATICS

The schematics for the COMPAQ Enhanced Color Graphics Board are shown in Figure 7-1. COMPAQ Computer Corporation does not guarantee the accuracy of the schematics. They are provided to aid in a general understanding of the board operation.



NOTES UNLESS OTHERWISE SPECIFIED:

1. ALL RESISTORS ARE EITHER 1/4W, 5% OR 1/8W, 1% AND ARE IN OHMS
2. ALL CAPACITORS ARE 50V AND ARE IN MICROFARADS.
3. R19 AND R26 ARE NOT INSTALLED.
4. U2, U16, AND U35 ARE INSTALLED WITH SOCKETS.

REVISIONS			APPROVALS			
REV	ECN NO.	DESCRIPTION OF CHANGE	DATE	DFTG	ENGR	REL.
A	6977	RELEASED FOR PRODUCTION				
B						

8
7
6
5
4
3
2
1
D
C
B
LS86
12
13
U10D
11
SPARE
3
4
7
8
13
14
17
18
1
11
2
5
6
9
12
15
16
19
U25
SPARES

Figure 7-1. COMPAQ Enhanced Color Graphics Board Schematics (Page 1 of 14)

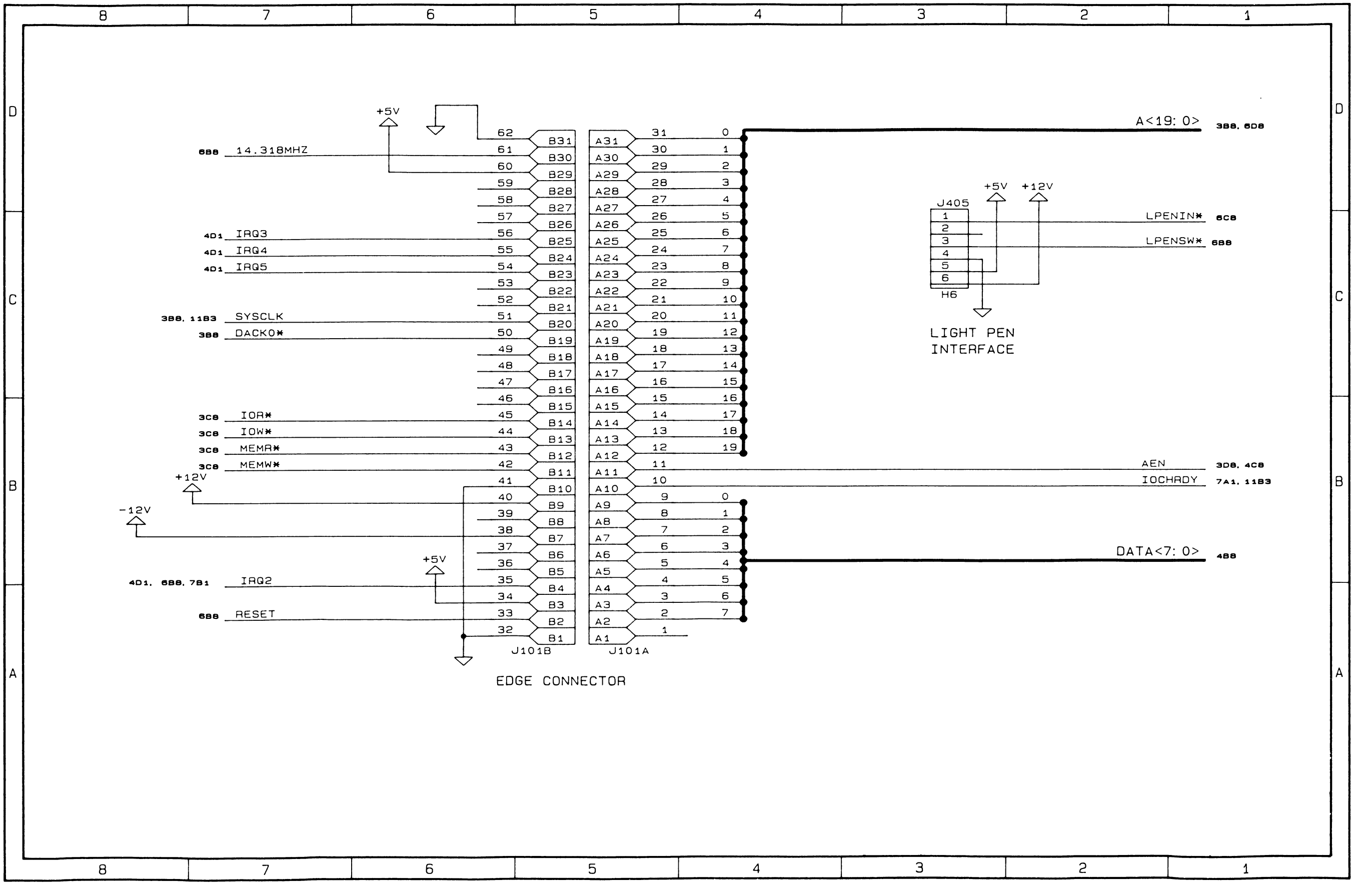

+5V
6B8 14.318MHZ
4D1 IRQ3
4D1 IRQ4
4D1 IRQ5
3B8, 11B3 SYSCLK
3B8 DACK0*
3C8 IOR*
3C8 IOW*
3C8 MEMR*
3C8 MEMW*
+12V
-12V
+5V
4D1, 6B8, 7B1 IRQ2
6B8 RESET
J101B
J101A
EDGE CONNECTOR
A<19: 0> 3B8, 6D8
J405
H6
+5V +12V
LPENIN* 6C8
LPENSW* 6B8
LIGHT PEN INTERFACE
AEN 3D8, 4C8
IOCHRDY 7A1, 11B3
DATA<7: 0> 4B8

Figure 7-1. COMPAQ Enhanced Color Graphics Board Schematics (Page 2 of 14)

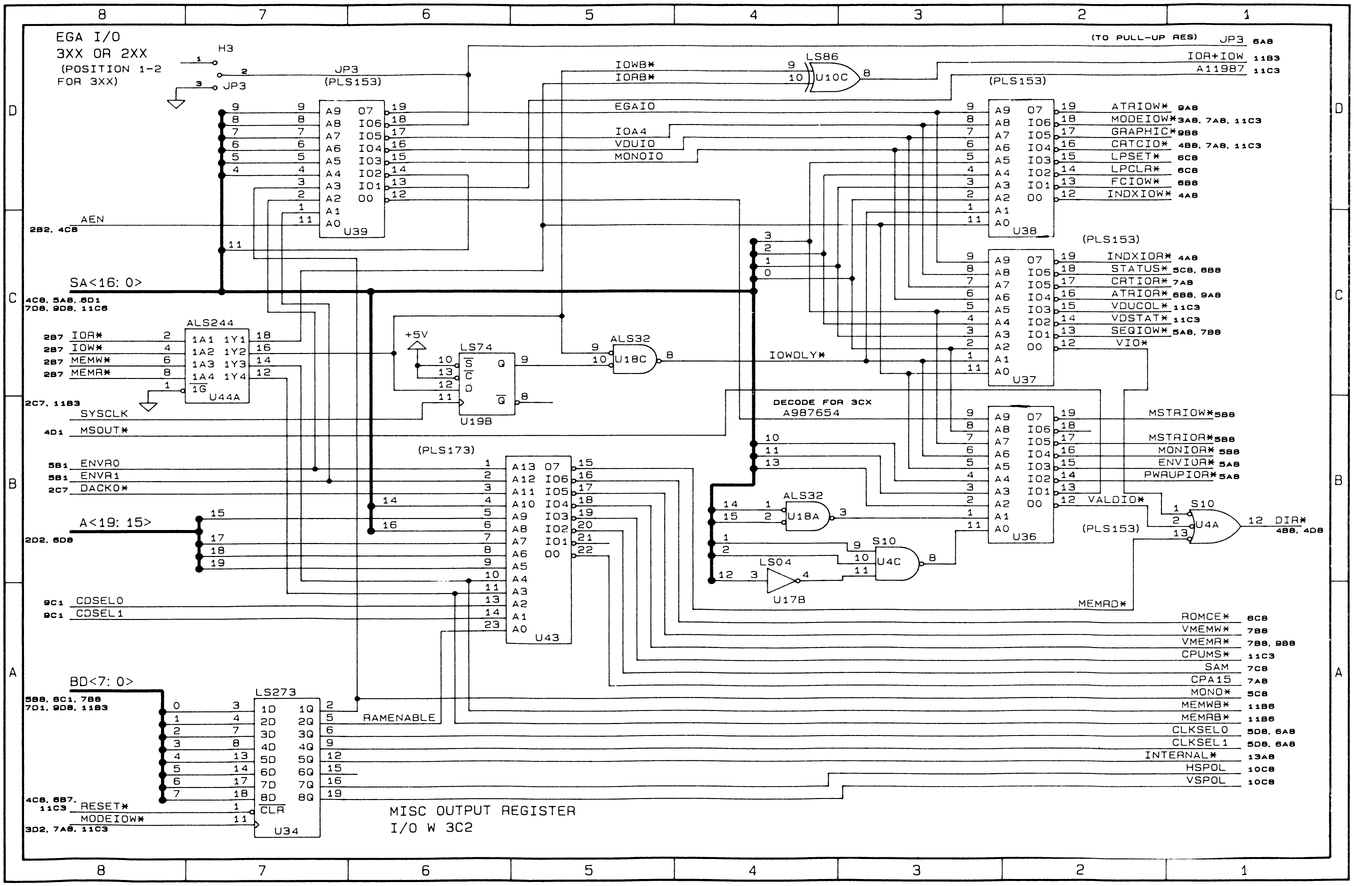

EGA I/O
3XX OR 2XX
(POSITION 1-2
FOR 3XX)
H3
JP3
JP3
(PLS153)
U39
AEN
2B2, 4C8
SA<16: 0>
4C8, 5A8, 8D1
7D8, 9D8, 11C8
ALS244
U44A
IOR*
IOW*
MEMW*
MEMR*
2C7, 11B3
SYSCLK
4D1
MSOUT*
5B1 ENVR0
5B1 ENVR1
2C7 DACK0*
A<19: 15>
2D2, 6D8
9C1 CDSEL0
9C1 CDSEL1
BD<7: 0>
5B8, 8C1, 7B8
7D1, 9D8, 11B3
4C8, 6B7, 11C3 RESET*
3D2, 7A8, 11C3 MODEIOW*
LS273
U34
RAMENABLE
MISC OUTPUT REGISTER
I/O W 3C2
+5V
LS74
U19B
(PLS173)
U43
IOWB*
IORB*
EGAIO
IOA4
VDUIO
MONOIO
LS86
U10C
ALS32
U18C
IOWDLY*
DECODE FOR 3CX
A987654
ALS32
U18A
S10
U4C
LS04
U17B
(TO PULL-UP RES) JP3 6A8
IOR+IOW 11B3
A11987 11C3
(PLS153)
U38
ATRIOW* 9A8
MODEIOW* 3A8, 7A8, 11C3
GRAPHIC* 9B8
CRTCIO* 4B8, 7A8, 11C3
LPSET* 6C8
LPCLR* 6C8
FCIOW* 6B8
INDXIOW* 4A8
(PLS153)
U37
INDXIOR* 4A8
STATUS* 5C8, 6B8
CRTIOR* 7A8
ATRIOR* 6B8, 9A8
VDUCOL* 11C3
VDSTAT* 11C3
SEQIOW* 5A8, 7B8
VIO*
U36
(PLS153)
MSTRIOW* 5B8
MSTRIOR* 5B8
MONIOR* 5B8
ENVIOR* 5A8
PWRUPIOR* 5A8
VALDIO*
S10
U4A
DIR* 4B8, 4D8
MEMRD*
ROMCE* 6C8
VMEMW* 7B8
VMEMR* 7B8, 9B8
CPUMS* 11C3
SAM 7C8
CPA15 7A8
MONO* 5C8
MEMWB* 11B6
MEMRB* 11B6
CLKSEL0 5D8, 6A8
CLKSEL1 5D8, 6A8
INTERNAL* 13A8
HSPOL 10C8
VSPOL 10C8

Figure 7-1. COMPAQ Enhanced Color Graphics Board Schematics (Page 3 of 14)

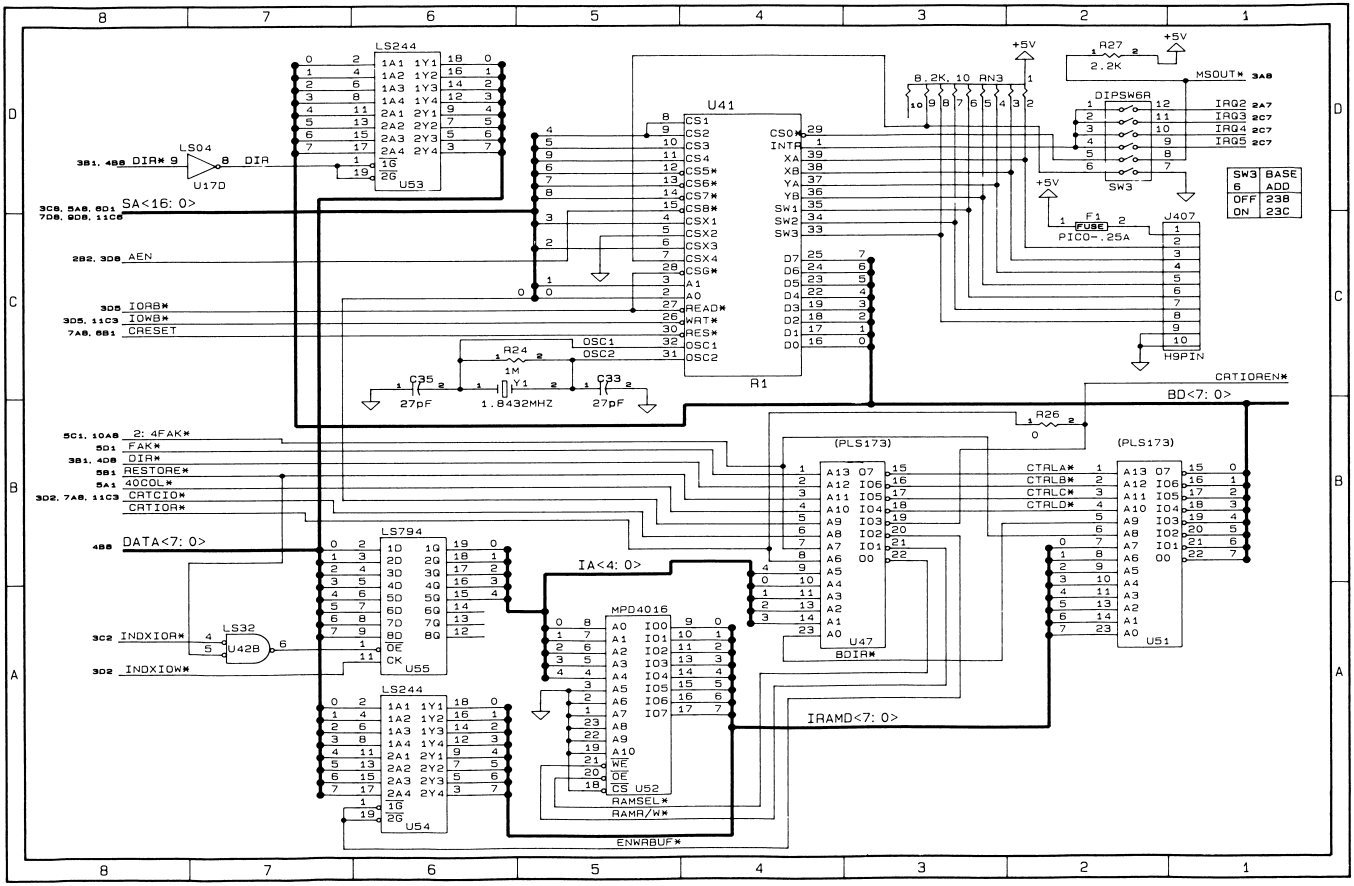

LS244
U53
LS04
U17D
DIR* 9
DIR
3B1, 4B8
SA<16: 0>
3C8, 5A8, 6D1
7D8, 9D8, 11C8
2B2, 3D8 AEN
3D5 IORB*
3D5, 11C3 IOWB*
7A8, 6B1 CRESET
U41
CS1
CS2
CS3
CS4
CS5*
CS6*
CS7*
CS8*
CSX1
CSX2
CSX3
CSX4
CSG*
A1
A0
READ*
WRT*
RES*
OSC1
OSC2
CS0*
INTR
XA
XB
YA
YB
SW1
SW2
SW3
D7
D6
D5
D4
D3
D2
D1
D0
R1
R24
1M
C35
27pF
Y1
1.8432MHZ
C33
27pF
+5V
R27
2.2K
8.2K, 10 RN3
MSOUT* 3A8
DIPSW6R
IRQ2 2A7
IRQ3 2C7
IRQ4 2C7
IRQ5 2C7
SW3
SW3 BASE 6 ADD
OFF 238
ON 23C
F1
FUSE
PICO-.25A
J407
H9PIN
CRTIOREN*
BD<7: 0>
R26
(PLS173)
U47
BDIR*
CTRLA*
CTRLB*
CTRLC*
CTRLD*
U51
5C1, 10A8 2: 4FAK*
5D1 FAK*
3B1, 4D8 DIR*
5B1 RESTORE*
5A1 40COL*
3D2, 7A8, 11C3 CRTCIO*
CRTIOR*
4B8 DATA<7: 0>
LS794
U55
OE
CK
3C2 INDXIOR*
LS32
U42B
3D2 INDXIOW*
LS244
U54
1G
2G
IA<4: 0>
MPD4016
U52
WE
OE
CS
RAMSEL*
RAMR/W*
ENWRBUF*
IRAMD<7: 0>
8
7
6
5
4
3
2
1
D
C
B
A

Figure 7-1. COMPAQ Enhanced Color Graphics Board Schematics (Page 4 of 14)

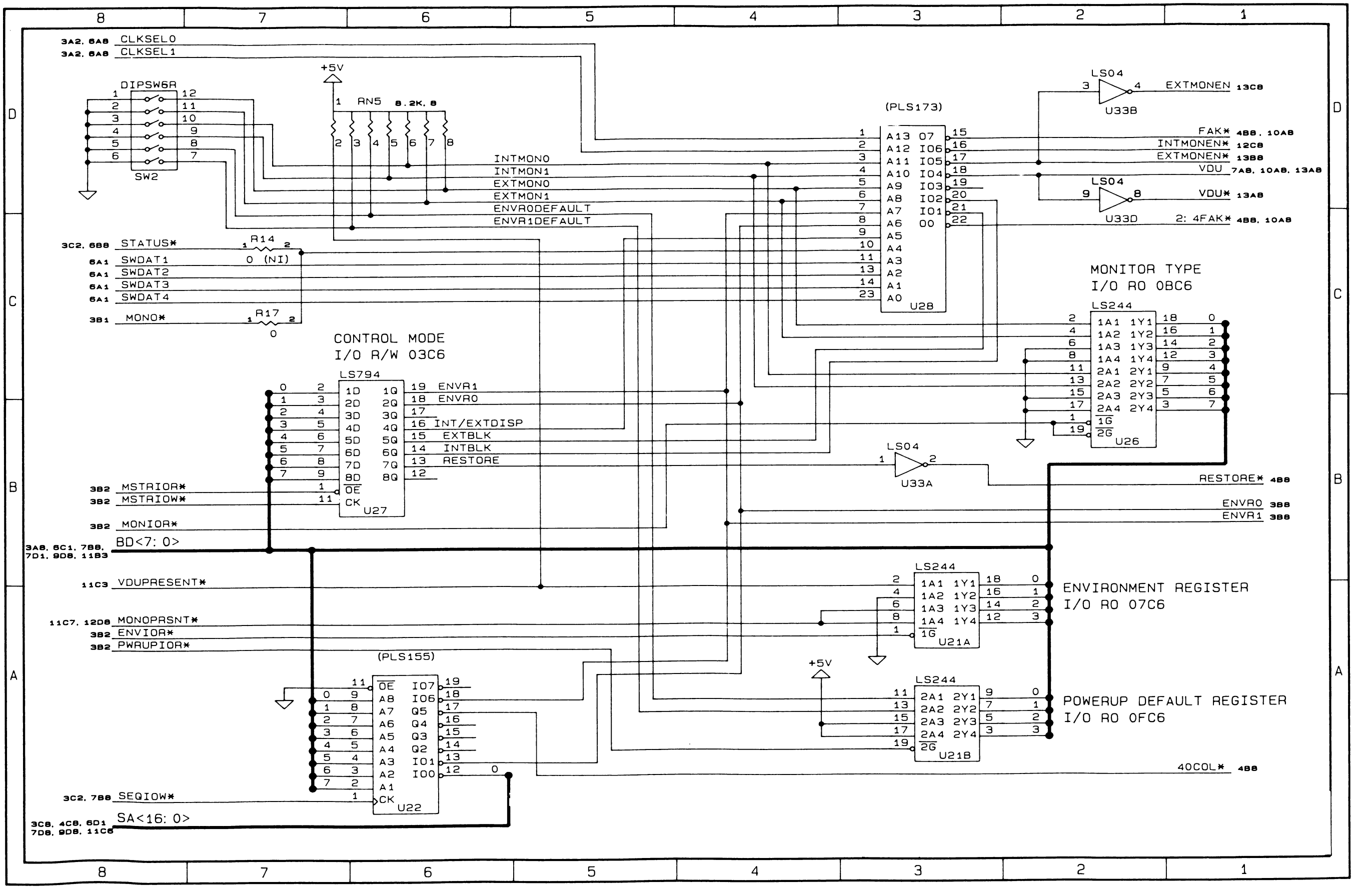

CLKSEL0
CLKSEL1
DIPSW6R
SW2
+5V
RN5 8.2K, 8
INTMON0
INTMON1
EXTMON0
EXTMON1
ENVR0DEFAULT
ENVR1DEFAULT
STATUS*
R14 0 (NI)
SWDAT1
SWDAT2
SWDAT3
SWDAT4
MONO*
R17 0
(PLS173)
U28
LS04
U33B
EXTMONEN
FAK*
INTMONEN*
EXTMONEN*
VDU
VDU*
U33D
2: 4FAK*
MONITOR TYPE
I/O R0 0BC6
LS244
U26
CONTROL MODE
I/O R/W 03C6
LS794
U27
ENVR1
ENVR0
INT/EXTDISP
EXTBLK
INTBLK
RESTORE
U33A
RESTORE*
ENVR0
ENVR1
MSTRIOR*
MSTRIOW*
MONIOR*
BD<7: 0>
VDUPRESENT*
ENVIRONMENT REGISTER
I/O R0 07C6
U21A
MONOPRSNT*
ENVIOR*
PWRUPIOR*
(PLS155)
U22
POWERUP DEFAULT REGISTER
I/O R0 0FC6
U21B
40COL*
SEQIOW*
SA<16: 0>

Figure 7-1. COMPAQ Enhanced Color Graphics Board Schematics (Page 5 of 14)

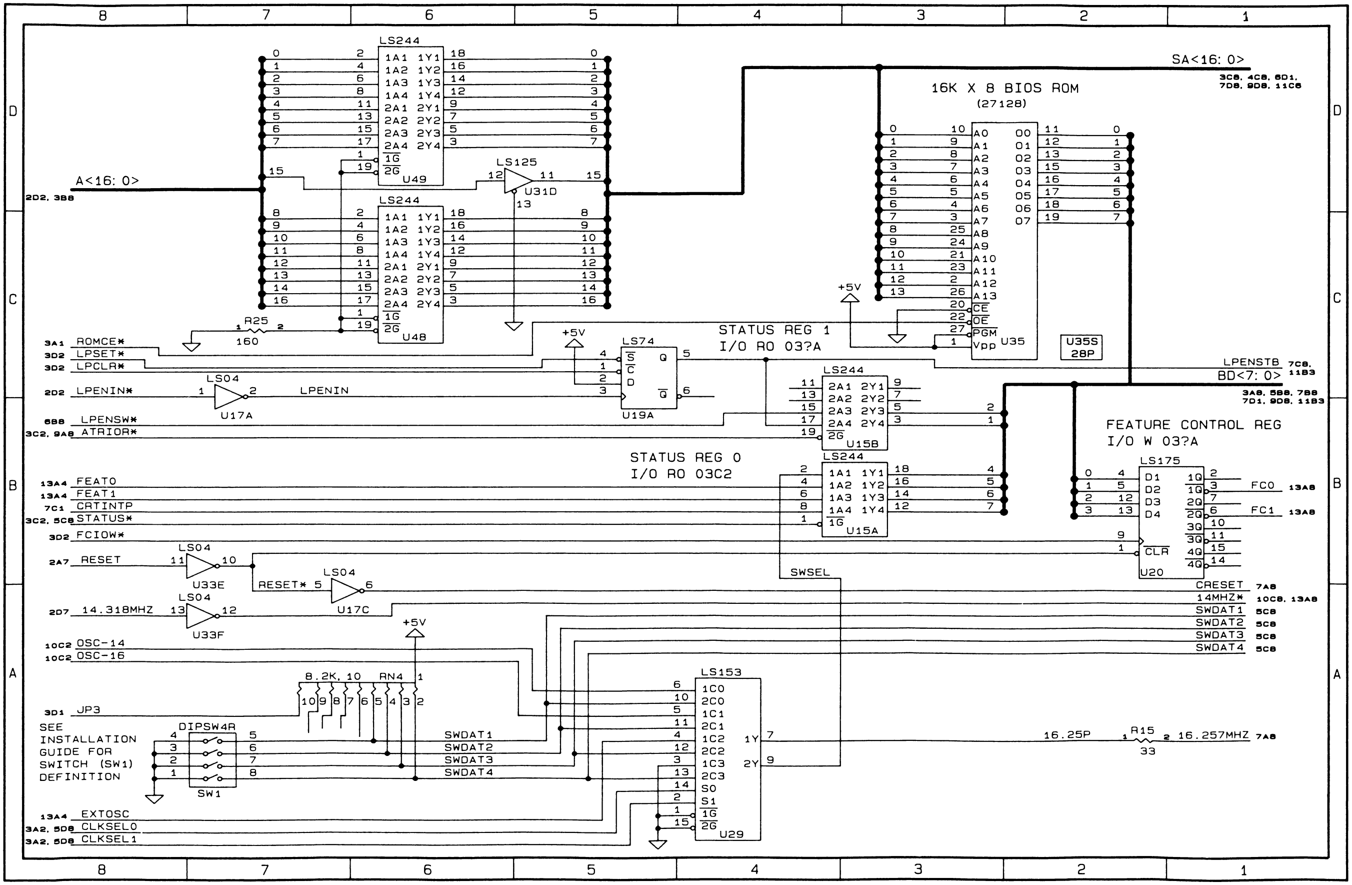

LS244
U49
U48
LS125
U31D
A<16: 0>
SA<16: 0>
16K X 8 BIOS ROM
(27128)
U35
U35S
28P
+5V
R25
160
ROMCE*
LPSET*
LPCLR*
LPENIN*
LS04
U17A
LPENIN
LS74
U19A
STATUS REG 1
I/O RO 03?A
LPENSTB
BD<7: 0>
LPENSW*
ATRIOR*
U15B
STATUS REG 0
I/O RO 03C2
U15A
FEATURE CONTROL REG
I/O W 03?A
LS175
U20
FEAT0
FEAT1
CRTINTP
STATUS*
FCIOW*
FC0
FC1
RESET
U33E
RESET*
U17C
SWSEL
CRESET
14MHZ*
14.318MHZ
U33F
SWDAT1
SWDAT2
SWDAT3
SWDAT4
OSC-14
OSC-16
8.2K, 10
RN4
JP3
SEE INSTALLATION GUIDE FOR SWITCH (SW1) DEFINITION
DIPSW4R
SW1
LS153
U29
16.25P
R15
33
16.257MHZ
EXTOSC
CLKSEL0
CLKSEL1

Figure 7-1. COMPAQ Enhanced Color Graphics Board Schematics (Page 6 of 14)

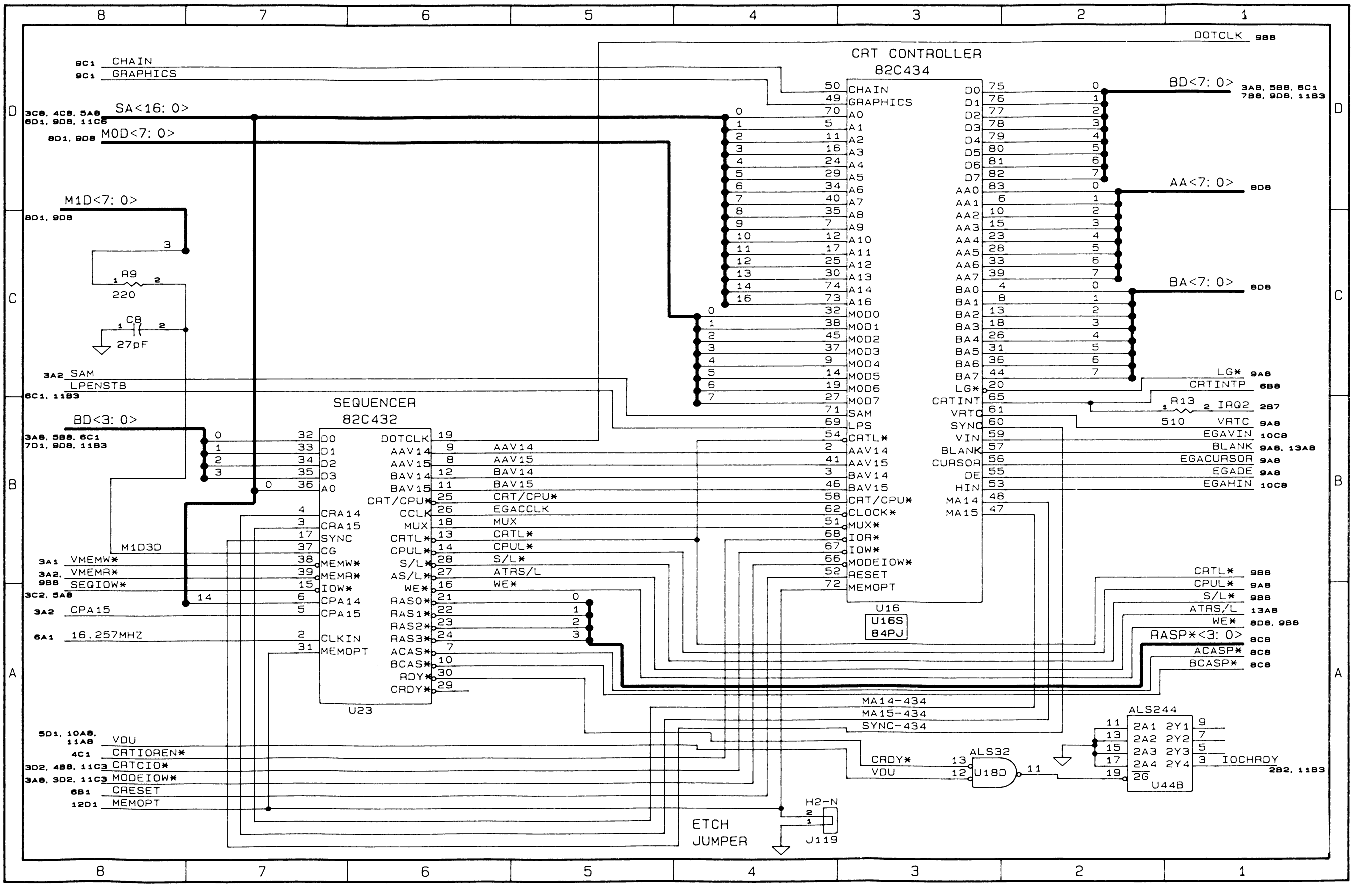

CRT CONTROLLER
82C434
U16
U16S
84PJ
SEQUENCER
82C432
U23
CHAIN
GRAPHICS
SA<16: 0>
MOD<7: 0>
M1D<7: 0>
R9
220
C8
27pF
SAM
LPENSTB
BD<3: 0>
M1D3D
VMEMW*
VMEMR*
SEQIOW*
CPA15
16.257MHZ
VDU
CRTIOREN*
CRTCIO*
MODEIOW*
CRESET
MEMOPT
DOTCLK
BD<7: 0>
AA<7: 0>
BA<7: 0>
LG*
CRTINTP
R13
510
IRQ2
VRTC
EGAVIN
BLANK
EGACURSOR
EGADE
EGAHIN
CRTL*
CPUL*
S/L*
ATRS/L
WE*
RASP*<3: 0>
ACASP*
BCASP*
AAV14
AAV15
BAV14
BAV15
CRT/CPU*
EGACCLK
MUX
MA14-434
MA15-434
SYNC-434
CRDY*
ALS32
U18D
ALS244
U44B
IOCHRDY
H2-N
J119
ETCH
JUMPER

Figure 7-1. COMPAQ Enhanced Color Graphics Board Schematics (Page 7 of 14)

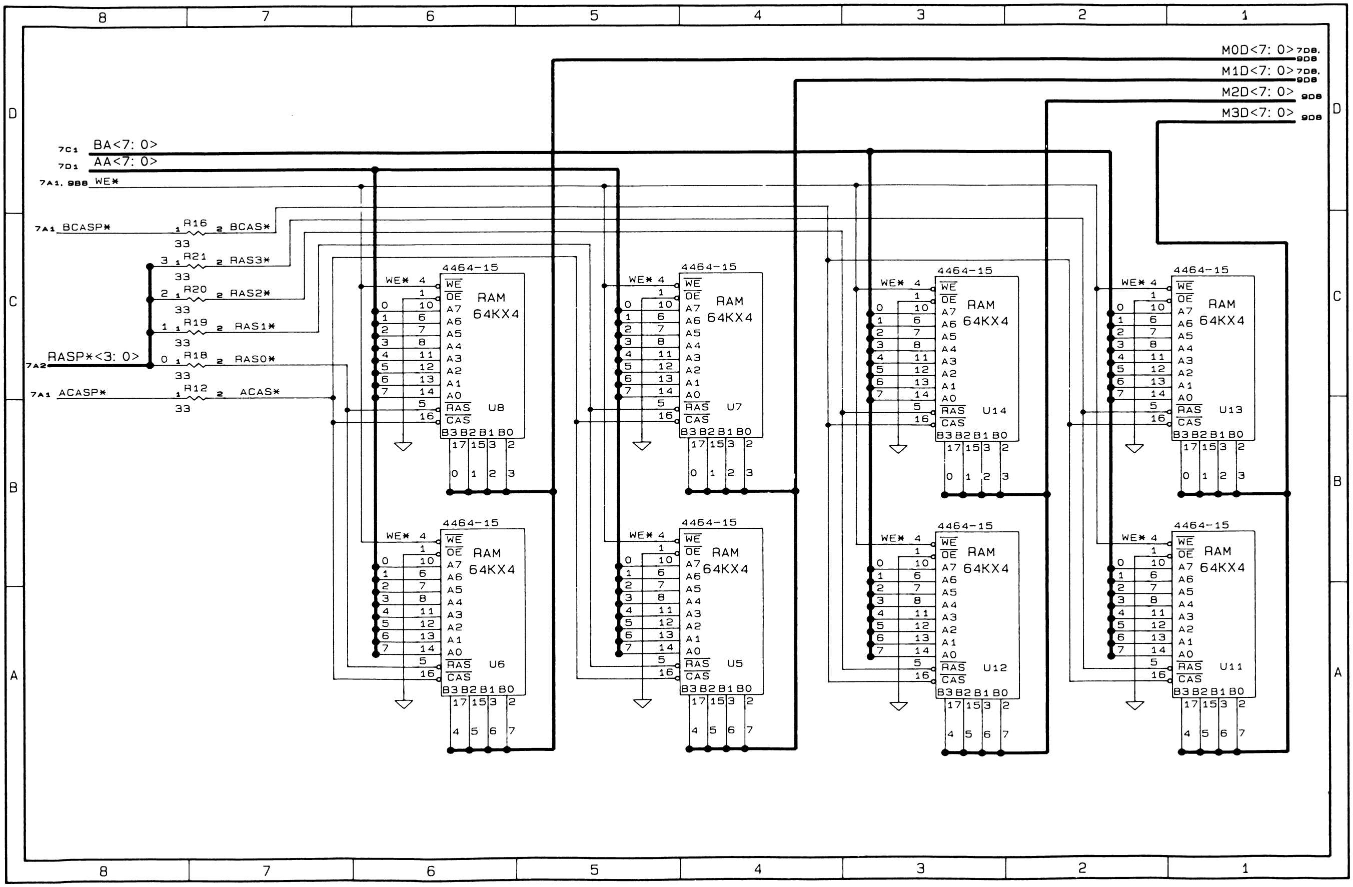

8
7
6
5
4
3
2
1
D
C
B
A
M0D<7: 0> 7D8, 9D8
M1D<7: 0> 7D8, 9D8
M2D<7: 0> 9D8
M3D<7: 0> 9D8
7C1 BA<7: 0>
7D1 AA<7: 0>
7A1, 9B8 WE*
7A1 BCASP*
R16 33
BCAS*
R21 33
RAS3*
R20 33
RAS2*
R19 33
RAS1*
7A2 RASP*<3: 0>
R18 33
RAS0*
7A1 ACASP*
R12 33
ACAS*
4464-15
RAM 64KX4
WE
OE
A7
A6
A5
A4
A3
A2
A1
A0
RAS
CAS
B3 B2 B1 B0
U8
U7
U14
U13
U6
U5
U12
U11

Figure 7-1. COMPAQ Enhanced Color Graphics Board Schematics (Page 8 of 14)

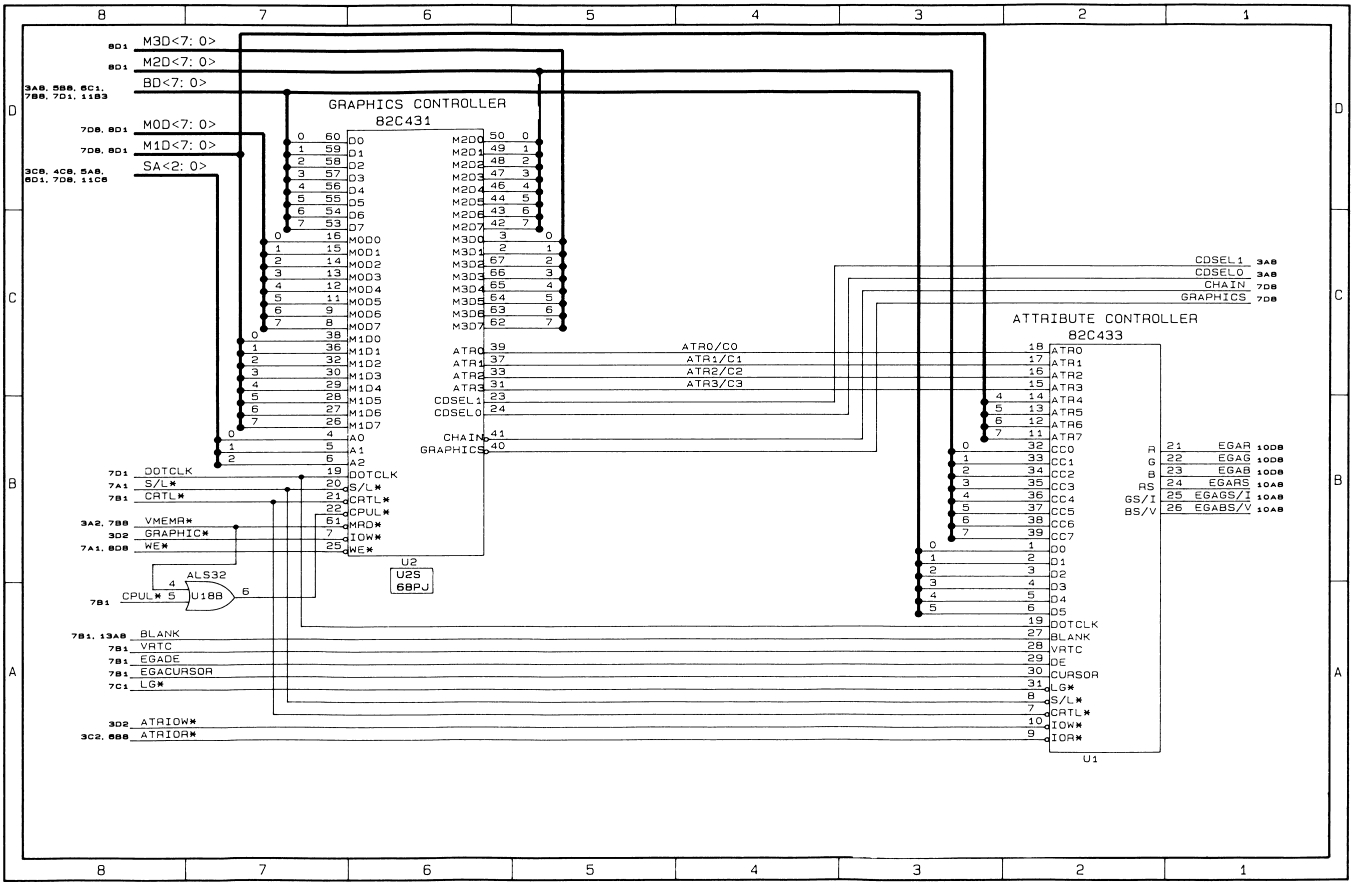

8
7
6
5
4
3
2
1
D
C
B
A
8D1 M3D<7: 0>
8D1 M2D<7: 0>
3A8, 5B8, 6C1, 7B8, 7D1, 11B3 BD<7: 0>
7D8, 8D1 M0D<7: 0>
7D8, 8D1 M1D<7: 0>
3C8, 4C8, 5A8, 6D1, 7D8, 11C6 SA<2: 0>
GRAPHICS CONTROLLER
82C431
D0
D1
D2
D3
D4
D5
D6
D7
MOD0
MOD1
MOD2
MOD3
MOD4
MOD5
MOD6
MOD7
M1D0
M1D1
M1D2
M1D3
M1D4
M1D5
M1D6
M1D7
A0
A1
A2
DOTCLK
S/L*
CRTL*
CPUL*
MRD*
IOW*
WE*
M2D0
M2D1
M2D2
M2D3
M2D4
M2D5
M2D6
M2D7
M3D0
M3D1
M3D2
M3D3
M3D4
M3D5
M3D6
M3D7
ATR0
ATR1
ATR2
ATR3
CDSEL1
CDSEL0
CHAIN
GRAPHICS
U2
U2S
68PJ
7D1 DOTCLK
7A1 S/L*
7B1 CRTL*
3A2, 7B8 VMEMR*
3D2 GRAPHIC*
7A1, 8D8 WE*
ALS32
U18B
7B1 CPUL*
7B1, 13A8 BLANK
7B1 VRTC
7B1 EGADE
7B1 EGACURSOR
7C1 LG*
3D2 ATRIOW*
3C2, 6B8 ATRIOR*
ATR0/C0
ATR1/C1
ATR2/C2
ATR3/C3
CDSEL1 3A8
CDSEL0 3A8
CHAIN 7D8
GRAPHICS 7D8
ATTRIBUTE CONTROLLER
82C433
ATR0
ATR1
ATR2
ATR3
ATR4
ATR5
ATR6
ATR7
CC0
CC1
CC2
CC3
CC4
CC5
CC6
CC7
D0
D1
D2
D3
D4
D5
DOTCLK
BLANK
VRTC
DE
CURSOR
LG*
S/L*
CRTL*
IOW*
IOR*
U1
R 21 EGAR 10D8
G 22 EGAG 10D8
B 23 EGAB 10D8
RS 24 EGARS 10A8
GS/I 25 EGAGS/I 10A8
BS/V 26 EGABS/V 10A8

Figure 7-1. COMPAQ Enhanced Color Graphics Board Schematics (Page 9 of 14)

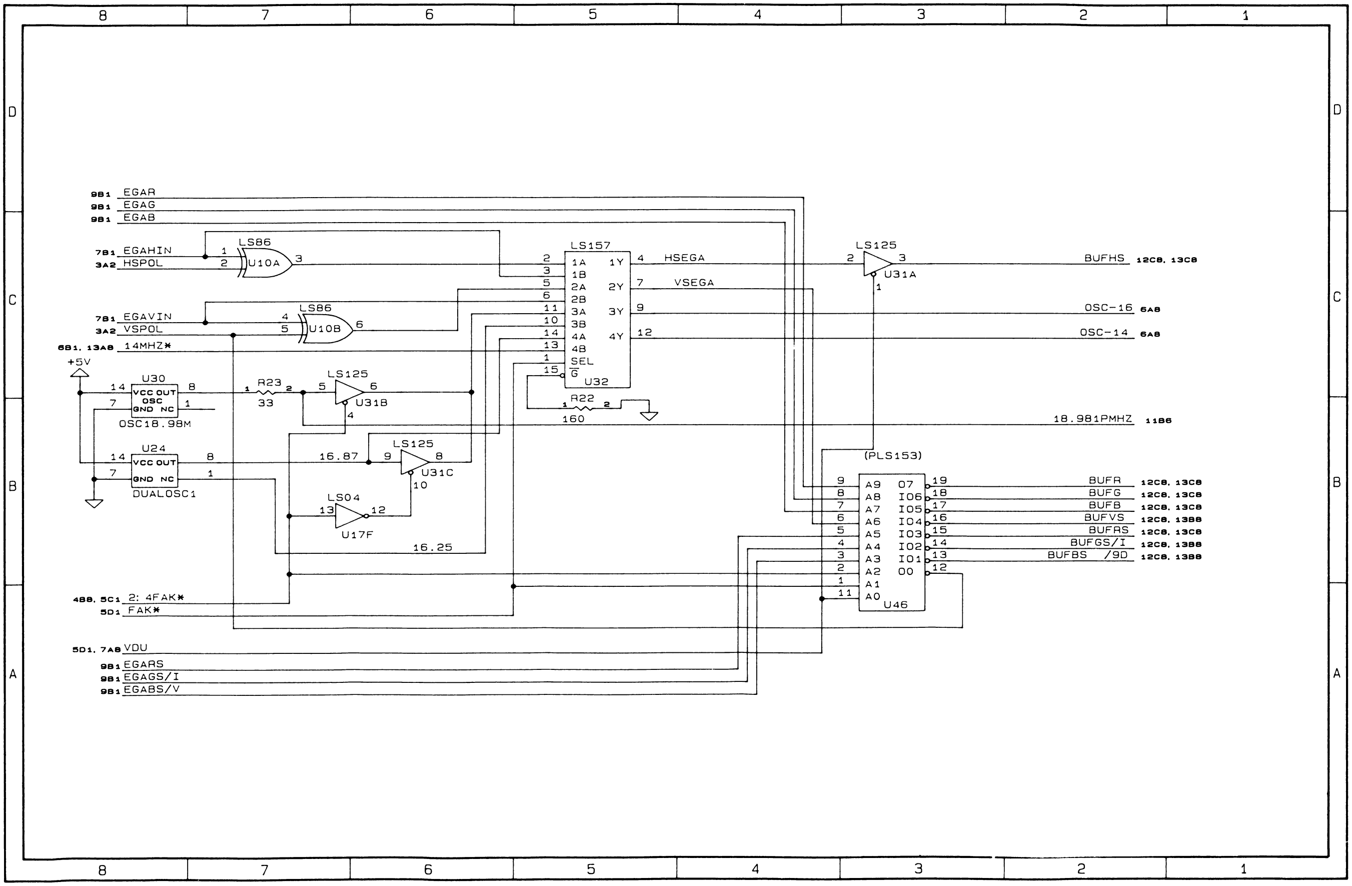

8
7
6
5
4
3
2
1
D
C
B
A
9B1 EGAR
9B1 EGAG
9B1 EGAB
7B1 EGAHIN
3A2 HSPOL
LS86
U10A
7B1 EGAVIN
3A2 VSPOL
LS86
U10B
6B1, 13A8 14MHZ*
+5V
U30
VCC OUT
OSC
GND NC
OSC18.98M
R23
33
LS125
U31B
U24
VCC OUT
GND NC
DUALOSC1
16.87
LS125
U31C
LS04
U17F
16.25
4B8, 5C1 2: 4FAK*
5D1 FAK*
5D1, 7A8 VDU
9B1 EGARS
9B1 EGAGS/I
9B1 EGABS/V
LS157
1A 1B 2A 2B 3A 3B 4A 4B SEL G
1Y 2Y 3Y 4Y
U32
R22
160
HSEGA
VSEGA
LS125
U31A
BUFHS 12C8, 13C8
OSC-16 6A8
OSC-14 6A8
18.981PMHZ 11B6
(PLS153)
A9 A8 A7 A6 A5 A4 A3 A2 A1 A0
O7 IO6 IO5 IO4 IO3 IO2 IO1 O0
U46
BUFR 12C8, 13C8
BUFG 12C8, 13C8
BUFB 12C8, 13C8
BUFVS 12C8, 13B8
BUFRS 12C8, 13C8
BUFGS/I 12C8, 13B8
BUFBS /9D 12C8, 13B8

Figure 7-1. COMPAQ Enhanced Color Graphics Board Schematics (Page 10 of 14)

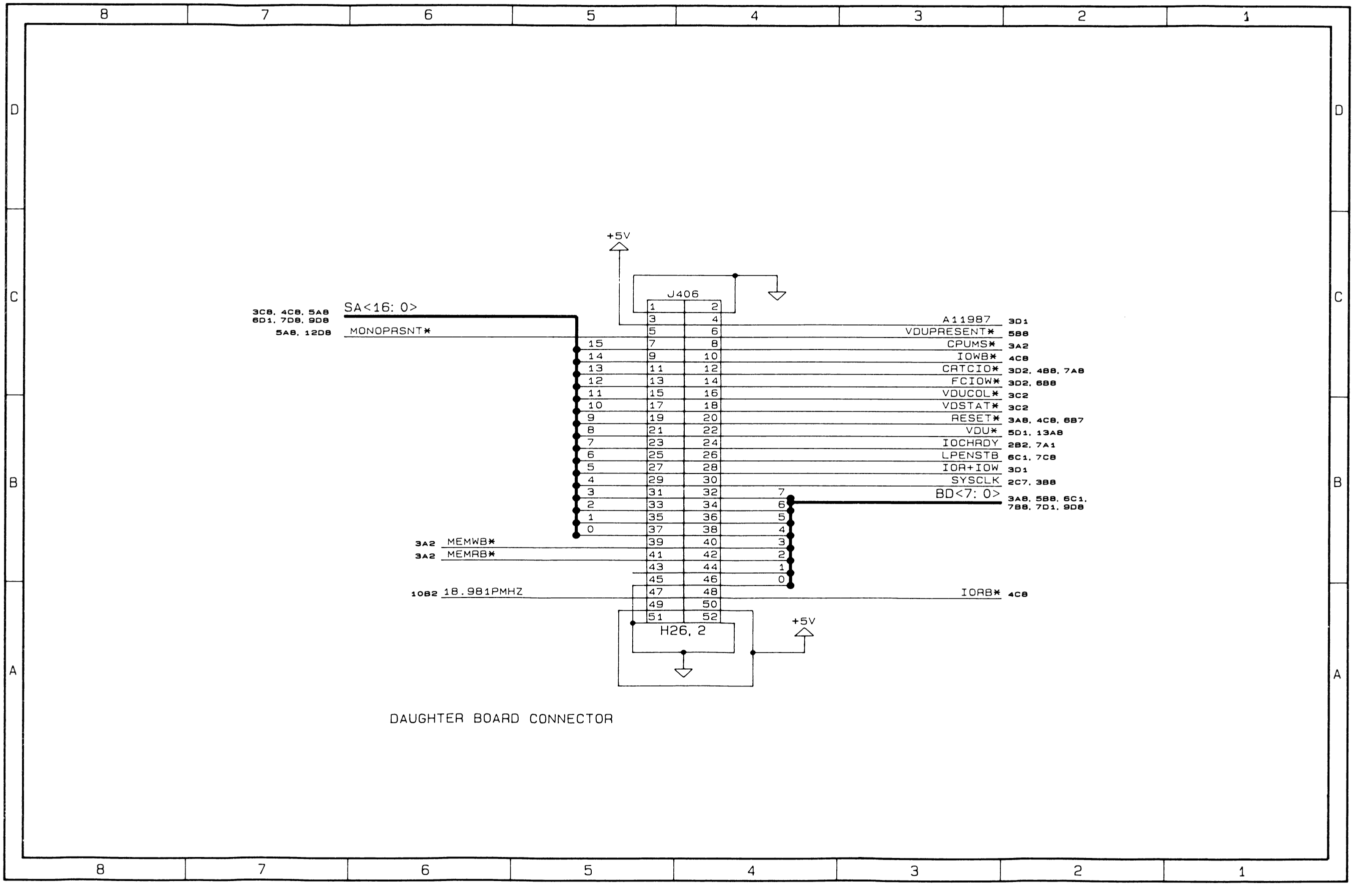

J406
+5V
SA<16: 0>
3C8, 4C8, 5A8
6D1, 7D8, 9D8
MONOPRSNT*
5A8, 12D8
A11987 3D1
VDUPRESENT* 5B8
CPUMS* 3A2
IOWB* 4C8
CRTCIO* 3D2, 4B8, 7A8
FCIOW* 3D2, 6B8
VDUCOL* 3C2
VDSTAT* 3C2
RESET* 3A8, 4C8, 6B7
VDU* 5D1, 13A8
IOCHRDY 2B2, 7A1
LPENSTB 6C1, 7C8
IOR+IOW 3D1
SYSCLK 2C7, 3B8
BD<7: 0>
3A8, 5B8, 6C1,
7B8, 7D1, 9D8
3A2 MEMWB*
3A2 MEMRB*
10B2 18.981PMHZ
IORB* 4C8
H26, 2
DAUGHTER BOARD CONNECTOR

Figure 7-1. COMPAQ Enhanced Color Graphics Board Schematics (Page 11 of 14)

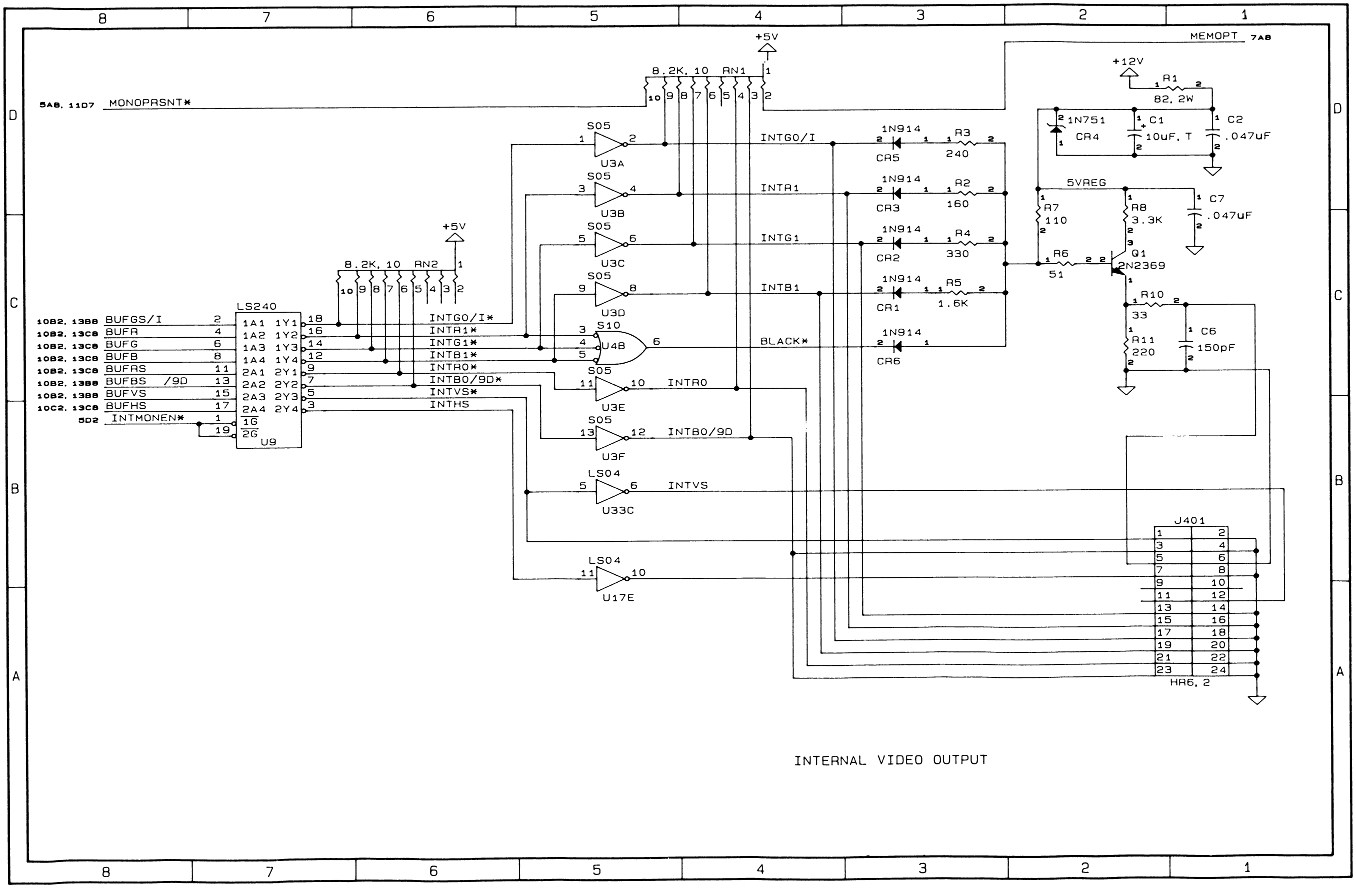

5A8, 11D7 MONOPRSNT*
MEMOPT 7A8
+5V
8.2K, 10 RN1
S05 U3A INTG0/I
S05 U3B INTR1
S05 U3C INTG1
S05 U3D INTB1
S10 U4B BLACK*
S05 U3E INTR0
S05 U3F INTB0/9D
LS04 U33C INTVS
LS04 U17E
8.2K, 10 RN2
LS240 U9
10B2, 13B8 BUFGS/I
10B2, 13C8 BUFR
10B2, 13C8 BUFG
10B2, 13C8 BUFB
10B2, 13C8 BUFRS
10B2, 13B8 BUFBS /9D
10B2, 13B8 BUFVS
10C2, 13C8 BUFHS
5D2 INTMONEN*
1A1 1A2 1A3 1A4 2A1 2A2 2A3 2A4 1G 2G
1Y1 1Y2 1Y3 1Y4 2Y1 2Y2 2Y3 2Y4
INTG0/I*
INTR1*
INTG1*
INTB1*
INTR0*
INTB0/9D*
INTVS*
INTHS
1N914 CR5 R3 240
1N914 CR3 R2 160
1N914 CR2 R4 330
1N914 CR1 R5 1.6K
1N914 CR6
+12V
R1 82, 2W
1N751 CR4
C1 10uF, T
C2 .047uF
5VREG
C7 .047uF
R7 110
R8 3.3K
R6 51
Q1 2N2369
R10 33
R11 220
C6 150pF
J401
HR6, 2
INTERNAL VIDEO OUTPUT
8 7 6 5 4 3 2 1
D C B A

Figure 7-1. COMPAQ Enhanced Color Graphics Board Schematics (Page 12 of 14)

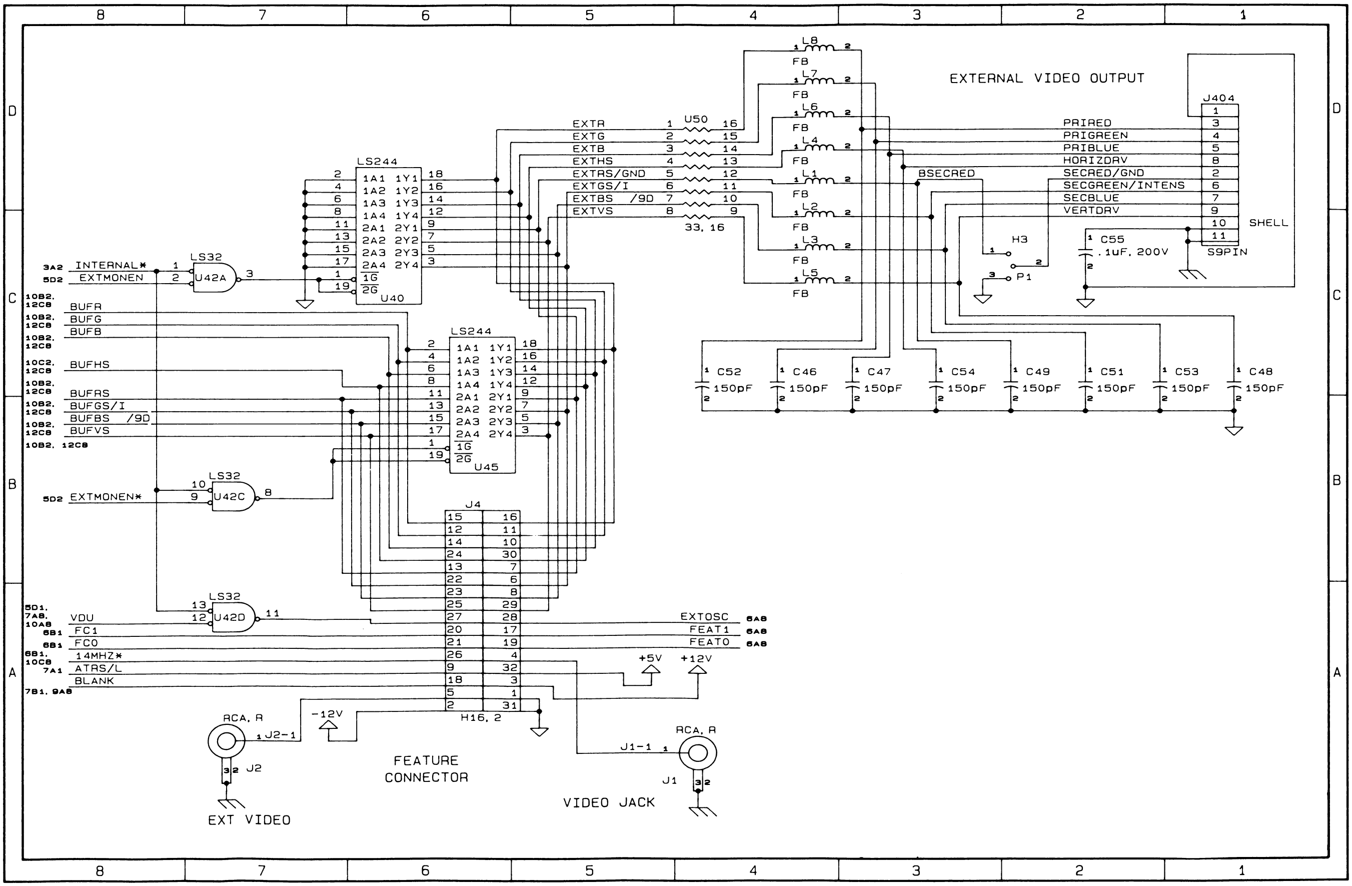

EXTERNAL VIDEO OUTPUT
J404
PRIRED
PRIGREEN
PRIBLUE
HORIZDRV
SECRED/GND
SECGREEN/INTENS
SECBLUE
VERTDRV
SHELL
S9PIN
BSECRED
H3
P1
C55 .1uF, 200V
L8 FB
L7 FB
L6 FB
L4 FB
L1 FB
L2 FB
L3 FB
L5 FB
U50
33, 16
EXTR
EXTG
EXTB
EXTHS
EXTRS/GND
EXTGS/I
EXTBS /9D
EXTVS
C52 150pF
C46 150pF
C47 150pF
C54 150pF
C49 150pF
C51 150pF
C53 150pF
C48 150pF
LS244
U40
1A1 1Y1
1A2 1Y2
1A3 1Y3
1A4 1Y4
2A1 2Y1
2A2 2Y2
2A3 2Y3
2A4 2Y4
1G
2G
LS244
U45
LS32
U42A
3A2 INTERNAL*
5D2 EXTMONEN
10B2, 12C8 BUFR
10B2, 12C8 BUFG
10B2, 12C8 BUFB
10C2, 12C8 BUFHS
10B2, 12C8 BUFRS
10B2, 12C8 BUFGS/I
10B2, 12C8 BUFBS /9D
10B2, 12C8 BUFVS
LS32
U42C
5D2 EXTMONEN*
LS32
U42D
5D1, 7A8, 10A8 VDU
6B1 FC1
6B1 FC0
6B1, 10C8 14MHZ*
7A1 ATRS/L
BLANK
7B1, 9A8
J4
H16, 2
EXTOSC 6A8
FEAT1 6A8
FEAT0 6A8
+5V
+12V
-12V
RCA, R
J2-1
J2
EXT VIDEO
FEATURE CONNECTOR
RCA, R
J1-1
J1
VIDEO JACK
8 7 6 5 4 3 2 1
D C B A

Figure 7-1. COMPAQ Enhanced Color Graphics Board Schematics (Page 13 of 14)

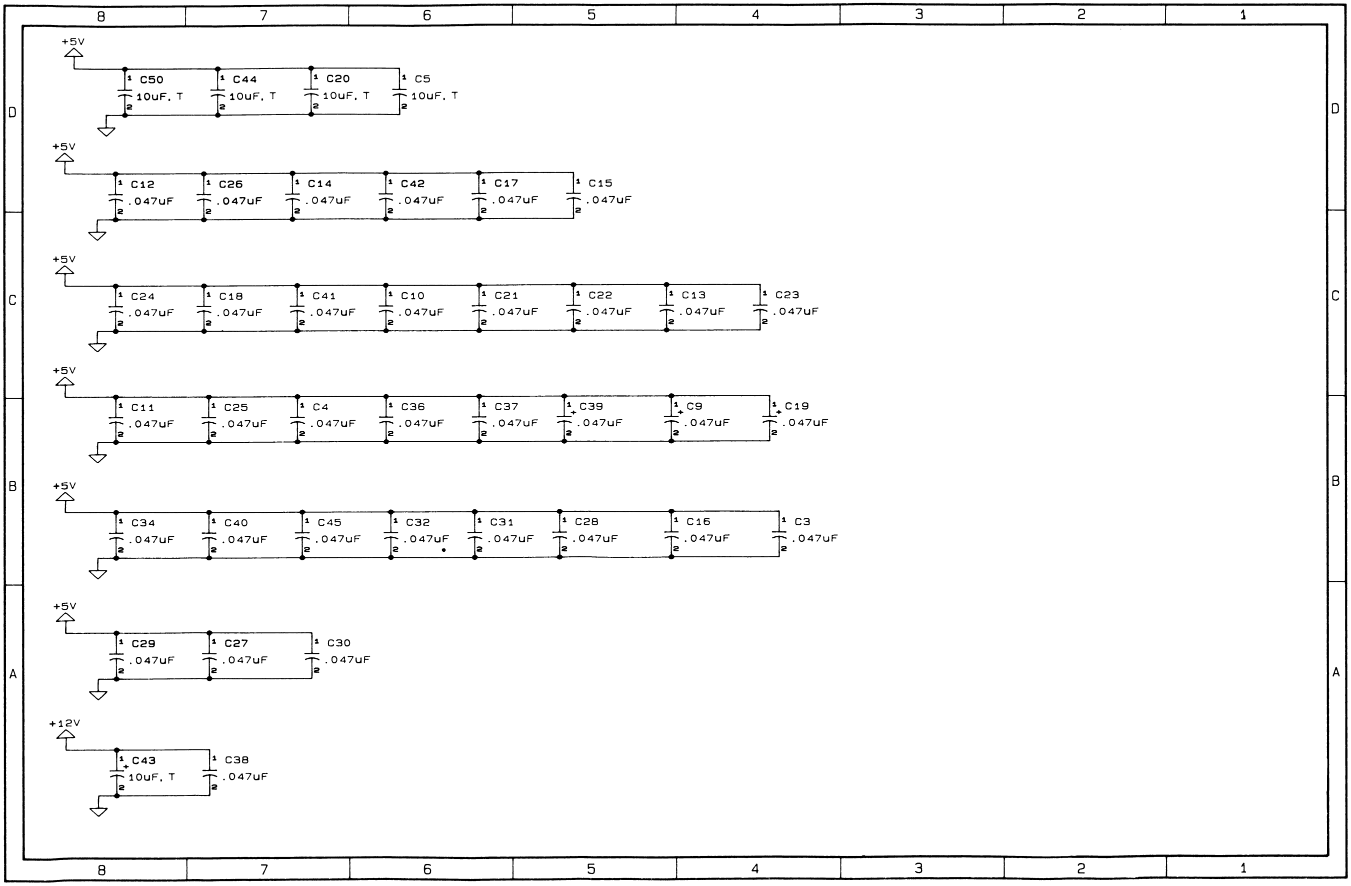

8
7
6
5
4
3
2
1
D
C
B
A
+5V
C50
10uF, T
C44
10uF, T
C20
10uF, T
C5
10uF, T
C12
.047uF
C26
.047uF
C14
.047uF
C42
.047uF
C17
.047uF
C15
.047uF
C24
C18
C41
C10
C21
C22
C13
C23
C11
C25
C4
C36
C37
C39
C9
C19
C34
C40
C45
C32
C31
C28
C16
C3
C29
C27
C30
+12V
C43
10uF, T
C38
.047uF

Figure 7-1. COMPAQ Enhanced Color Graphics Board Schematics (Page 14 of 14)

Appendix A
COMPAQ COLOR MONITOR

A.1 INTRODUCTION

The COMPAQ Color Monitor, shown in Figure A-1, is a dual-frequency color monitor that operates the COMPAQ Enhanced Color Graphics Board and the COMPAQ Video Display Controller Board. The COMPAQ Enhanced Color Graphics Board allows the monitor to display as many as 64 colors, using a 6-bit color definition.

The dual-frequency feature allows the monitor to function at two horizontal frequencies. The 15.7-kilohertz (kHz) horizontal sweep frequency generates 200 active (or visible) scan lines per vertical frame. The 21.8-kHz horizontal sweep frequency generates 350 active (or visible) scan lines per vertical frame.

Figure A-2 is a functional block diagram of the COMPAQ Color Monitor.

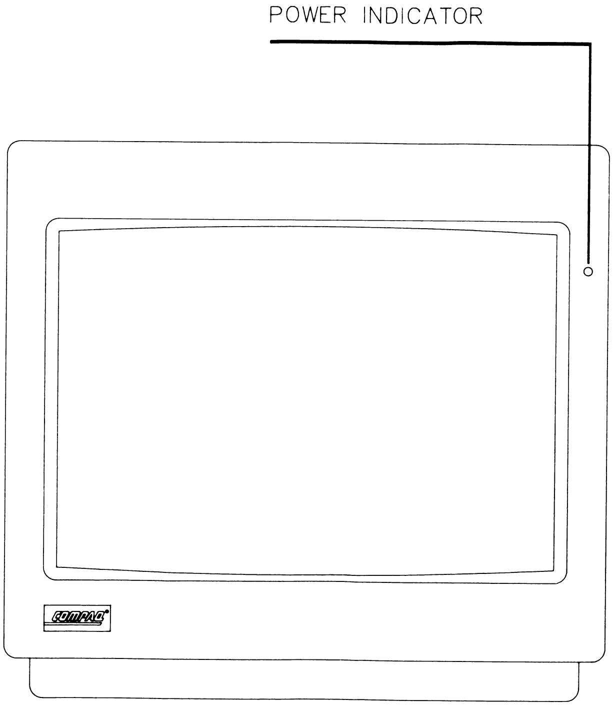


Figure A-1. COMPAQ Color Monitor

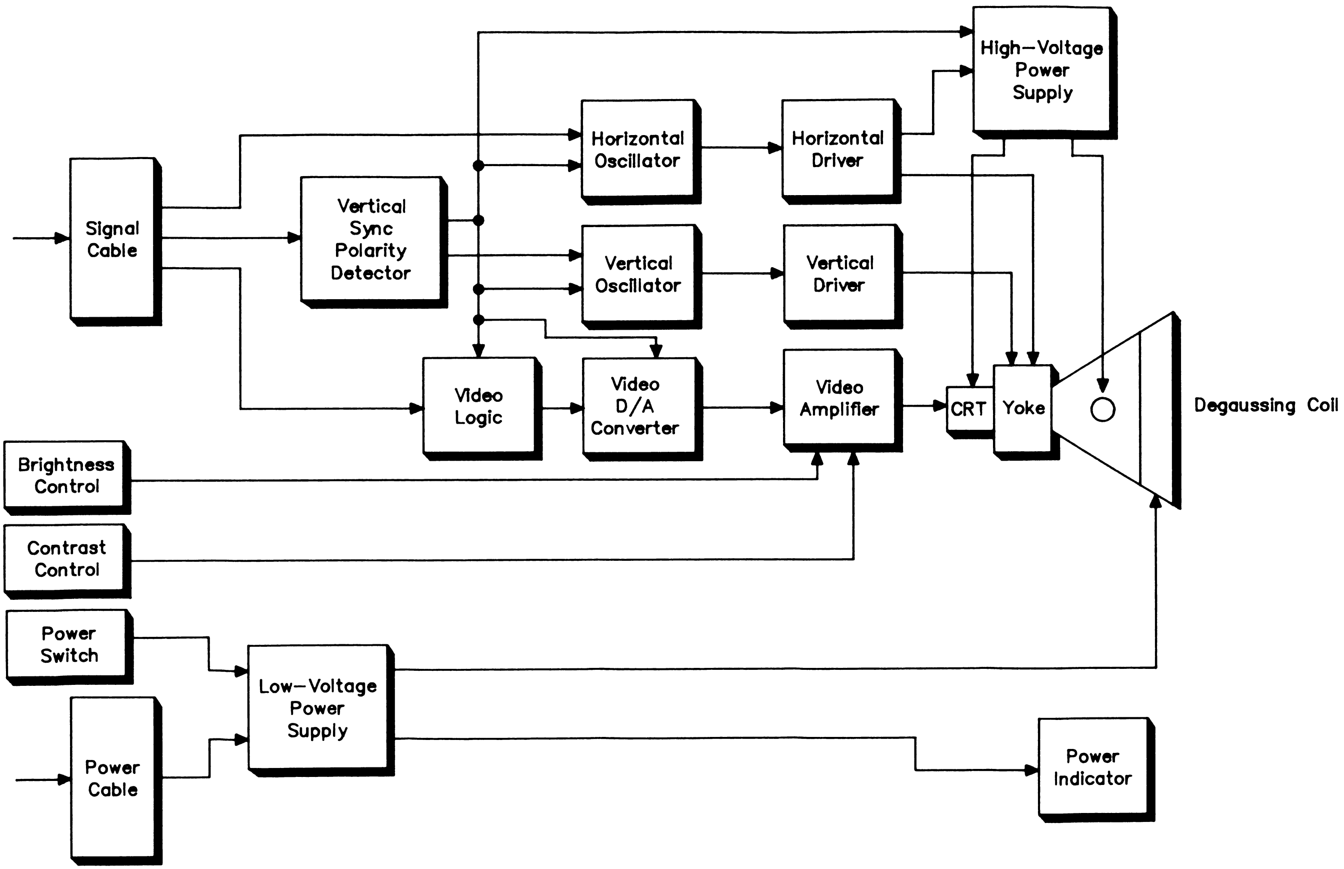

High–Voltage Power Supply
Horizontal Oscillator
Horizontal Driver
Signal Cable
Vertical Sync Polarity Detector
Vertical Oscillator
Vertical Driver
Video Logic
Video D/A Converter
Video Amplifier
CRT
Yoke
Degaussing Coil
Brightness Control
Contrast Control
Power Switch
Low–Voltage Power Supply
Power Cable
Power Indicator

Figure A-2. COMPAQ Color Monitor Functional Block Diagram

A.2 SPECIFICATIONS

Table A-1 lists the COMPAQ Color Monitor specifications.

Table A-1. COMPAQ Color Monitor Specifications

Monitor:	
Height	12 in. (30.4 cm)
Depth	15 in. (38.0 cm)
Width	13 in. (35.4 cm)
Weight	28 lb (12.8 kg)
Display:	
640 pixels x 350 scan lines (64 colors)	
640 pixels x 200 scan lines (16 colors)	
13-in. screen (diagonal)	
90-degree deflection	
Signal Cable:	
Length	60 in. (1.5 m)
Interface	RGBI/RrGgBb using shielded 9-pin subminiature D-type connector
Power:	
Cable Length	60 in. (1.5 m)
Cable Connector	NEMA 5-15P (parallel blade plug, (North American units only)
Voltage:	
North American units	120 VAC (60 Hz)
International units	220-240 VAC (50 to 60 Hz)
Power Consumption	100 W (maximum) 75 W (typical)
Environmental Requirements:	
Temperature:	
Operating	50°F to 104°F (10°C to 40°C)
Nonoperating	14°F to 131°F (-10°C to 55°C)
Humidity	10% to 95% (noncondensing)
Altitude	Mean sea level to 7000 ft. (2 100 m)

A.3 INSTALLATION CONSIDERATIONS

Adjustable Tilt Bar

The bottom of the monitor has a tilt bar that allows the monitor to be viewed in three positions: level and 5-degree or 10-degree-angled views.

To adjust the tilt bar, gently set the monitor on its left side, push the spring-loaded tilt bar down and set it as desired. Then release the tilt bar and lock it into place.

Video Display Controller Board Compatibility

Connect the COMPAQ Color Monitor only to a video display controller board having standard RGBI or RrGgBb video outputs with a horizontal frequency output of 15.7 kHz or 21.8 kHz. The COMPAQ Color Monitor works with either the COMPAQ Enhanced Color Graphics Board or the COMPAQ Video Display Controller Board. When the monitor is connected to a video display controller board with incompatible video output signals, the display image, shape, and position may be distorted.

Note: The switch settings on the video display controller board may need to be changed before the COMPAQ Color Monitor functions correctly. (See your video display controller board operations guide for the correct switch settings for the COMPAQ Color Monitor.)

Connecting AC Power

For the North American Version of the COMPAQ Color Monitor, the alternating-current (AC) power cable is an integral part of the monitor. To connect the monitor to an AC power source, plug the power cable into a standard 120-VAC (60-Hz) receptacle.

For the international version of the COMPAQ Color Monitor, the AC power cable is a separate cable that must be plugged in to both the monitor and the AC receptacle. Connect the "monitor end" of the power cable to the connector on the back of the monitor. Plug the "AC end" of the power cable into the AC receptacle. International units operate on 220- to 240-VAC, 50- to 60-Hz power.

Controls

Figure A-3 shows the following COMPAQ Color Monitor controls:

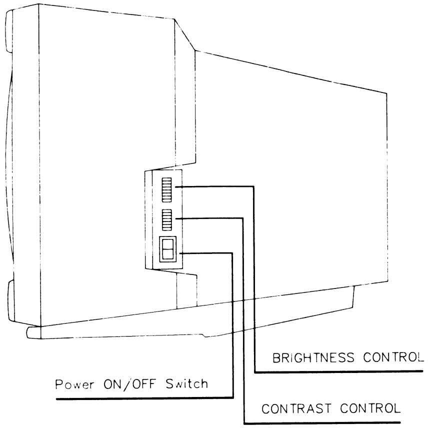
- Power ON/OFF Switch. The Power ON/OFF Switch turns on and off the monitor's AC power. Push the top of the "rocker-type" switch to turn on the monitor. Push the bottom of the switch to turn off the monitor.

- Contrast Control. The Contrast Control adjusts the difference in the degree of brightness among the colors. To increase the contrast, rotate the Contrast Control downward. To decrease the contrast, rotate the Contrast Control upward. The contrast control is preset in the 350-scan line mode.
- Brightness Control. The Brightness Control adjusts the brightness of the display. To increase the brightness, rotate the Brightness Control downward. To decrease the brightness, rotate the Brightness Control upward.
- Vertical-Height, High-Frequency Mode Control (V-Height-H). The V-Height-H Control adjusts the screen size for the 350-scan line mode. Adjust this control when you are in the 350-scan line mode until the display height is suitable.
- Vertical-Height, Low-Frequency Mode Control (V-Height-L). The V-Height-L Control adjusts the screen size for the 200-scan line mode. Adjust this control when you are in the 200-scan line mode until the display height is suitable.

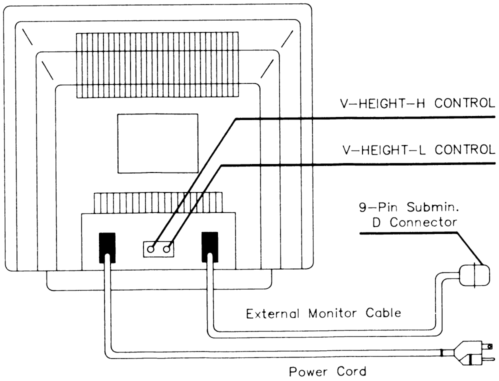
WARNING

DO NOT ATTEMPT TO ADJUST ANY INTERNAL CONTROLS:

1. **REMOVING THE COVER OF THE MONITOR EXPOSES YOU TO DANGEROUS VOLTAGES.**
2. **IMPROPER ADJUSTMENT OF THE INTERNAL CONTROLS CAN PERMANENTLY DAMAGE THE MONITOR.**
3. **ATTEMPTING TO ADJUST THE INTERNAL CONTROLS VOIDS THE WARRANTY ON THE MONITOR.**


BRIGHTNESS CONTROL
Power ON/OFF Switch
CONTRAST CONTROL

Right-Side View


V-HEIGHT-H CONTROL
V-HEIGHT-L CONTROL
9-Pin Submin.
D Connector
External Monitor Cable
Power Cord

Rear View

Figure A-3. COMPAQ Color Monitor Controls

A.4 SELF-TEST

The COMPAQ Color Monitor has a self-test system. To perform this self-test:

1. Turn off the monitor and disconnect the monitor signal cable from the computer.
2. Turn on the monitor and look at the screen. After warm-up, the screen color will be white if the monitor is working properly .

If the screen is dark or any color other than white, contact your Authorized COMPAQ Computer Dealer for service.

A.5 CONNECTORS

The COMPAQ Color Monitor has one signal connector, which is shown in Figure A-4. The monitor uses the polarity of the vertical sync (VS) signal coming from the video display controller board to establish 200- or 350-scan line operation. When the VS signal is active high (+ polarity), the monitor operates in the 200-scan line mode. When the VS signal is active low (- polarity), the monitor operates in the 350 scan-line mode.

Table A-2 lists the monitor connector signal descriptions.

Table A-2. Monitor Connector Signal Descriptions

Pin	200-Scan Line Description	350-Scan Line Description
1	Signal Ground	Signal Ground
2	Not used	r-secondary red
3	R-red	R-primary red
4	G-green	G-primary green
5	B-blue	B-primary blue
6	I-intensity	g-secondary green
7	Not used	b-secondary blue
8	HS-horizontal sync	HS-horizontal sync
9	VS-vertical sync	VS-vertical sync

Note: All signals are active-high TTL signals, except the VS signal. Polarity of VS determines the number of scan lines.

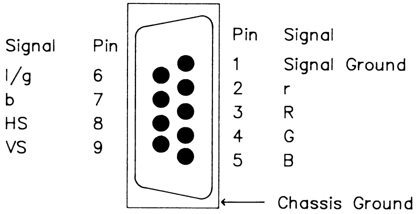
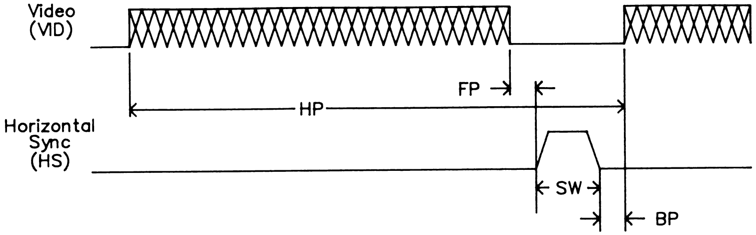


Figure A-4. COMPAQ Color Monitor Signal Connector

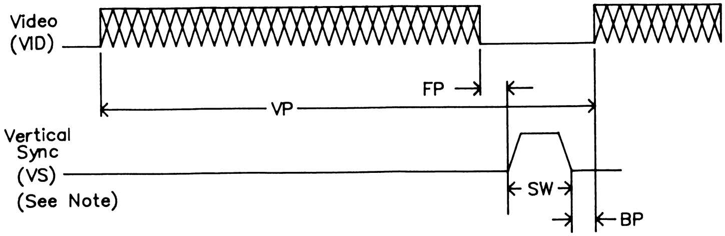
A.6 SIGNAL TIMING WAVEFORMS

Figures A-5 and A-6 show the timing parameters for the COMPAQ Color Monitor.


Video
(VID)
FP
HP
Horizontal
Sync
(HS)
SW
BP

Symbol	Parameter	350 Scan lines	200 Scan lines
HP	Horizontal Period	45.8 μs	63.7 μs
FP	Front Porch	0 μs	6.7 μs
SW	Sync Width	4.9 μs	4.5 μs
BP	Back Porch	1.5 μs	7.8 μs

Figure A-5. COMPAQ Color Monitor Horizontal Timing


Video
(VID)
FP
VP
Vertical
Sync
(VS)
(See Note)
SW
BP

Symbol	Parameter	350 Scan lines	200 Scan lines
VP	Vertical Period	16.7 ms	16.7 ms
FP	Front Porch	0 ms	1.5 ms
SW	Sync Width	0.6 ms	0.2 ms
BP	Back Porch	0.1 ms	2.2 ms

Note: VS is active low for the 350–scan line mode.
VS is active high for the 200–scan line mode.

Figure A-6. COMPAQ Color Monitor Vertical Timing

Appendix B
PROGRAMMING EXAMPLES

This appendix provides ten examples (Sections B.1 through B.10) of assembly-language programming routines that demonstrate software-accessible features of the COMPAQ Enhanced Color Graphics Board. Another section (Section B.11) consists of eight individual subroutines that are called into the ten examples. The sections and their routines are:

- B.1, Clearing the Screen
- B.2, Loading the Character Font
- B.3, Showing a Split Screen
- B.4, Logical Windowing
- B.5, Smooth Scrolling and Panning
- B.6, Using the Vertical Blanking Interrupt
- B.7, Programming the Color Palette
- B.8, Modifying Video Parameters
- B.9, Making Special Fonts Resident
- B.10, Switching Between Color Text and Monochrome Text Display Modes
- B.11, Miscellaneous Routines

These routines utilize both the video BIOS INT 10h and direct register programming of the hardware. Instruction sequences are processor speed-independent and execute on 8088-, 8086-, 80286-, and 80386-based systems. Examples given are not intended to be stand-alone programs, but rather subroutines callable from a higher level routine.

B.1 CLEARING THE SCREEN

This example shows how to clear the screen in a manner independent of the display controller used or its operating mode (25 or 43 line, etc). Use of the SCROLL UP BIOS function is one of the fastest ways to erase the screen. It also avoids the visual side effects seen on the monitor that can result from a disturbance in the sync signal when the CRT controller is reprogrammed (that is, a mode change).

```
Microsoft (R) Macro Assembler  Version 4.00                 9/9/86 12:00:00

Clear_Screen    Sample Clear Screen Routine                 Page     1-1


                                        page    30,120
                                        title   Clear_Screen    Sample Clear Screen Routine
                                ;****************************************************a*************************
                                ;
                                ;  Module Functional Description:
                                ;
                                ;               This module contains a routine to clear the screen
                                ;       of any COMPAQ video adapter. This clear screen uses the Scroll
                                ;       Up BIOS call and will work properly in any video mode.
                                ;
                                ;******************************************************************************
                                        public  Clear_Screen            ; Generic clear screen routine
                                ;
                                ;  LOCAL Equates
                                ;
 = 0040                         BIOS_SEG        equ     0040h           ; BIOS RAM data area segment (400h abs.)
 = 0010                         VIDEO           equ     10h             ; BIOS video interrupt number
 = 0006                         SCROLL_UP       equ     06h             ; Video BIOS scroll up function
 = 000F                         GET_MODE        equ     0Fh             ; Get current video mode function code
 = 007F                         MODE_MASK       equ     7Fh             ; Current video mode mask value
 = 0003                         TEXT_MODE       equ     03h             ; Maximum color/graphics text mode
 = 0007                         MONO_MODE       equ     07h             ; Monochrome text mode
 = 0007                         TEXT_ATTR       equ     07h             ; Text mode attribute for the scroll
 = 0000                         GRAPHICS_ATTR   equ     00h             ; Graphics mode attribute for scroll
 = 0018                         DEFAULT_ROWS    equ     25 - 1          ; Default number of rows on the screen
 = 0084                         V_ROWS          equ     BYTE PTR 84h    ; Offset of number of rows on screen
```

```
Microsoft (R) Macro Assembler  Version 4.00             9/9/86 12:00:00

Clear_Screen    Sample Clear Screen Routine             Page     1-2


                                  ;
                                  ;  Define the code segment
                                  ;
 0000                             code    segment byte public 'CODE'      ; Standard code segment
                                          assume  cs:code, ds:nothing
```

```
Microsoft (R) Macro Assembler  Version 4.00                  9/9/86 12:00:00

Clear_Screen    Sample Clear Screen Routine                  Page     1-3


                                         page
                                 ;*******************************************************************************
                                 ;
                                 ;       Clear_Screen()
                                 ;
                                 ;       Registers on Entry:
                                 ;
                                 ;               None
                                 ;
                                 ;       Registers on Exit:
                                 ;
                                 ;               None
                                 ;
                                 ;*******************************************************************************
 0000                            Clear_Screen    Proc    Near            ; Clear screen procedure
 0000  50                                push    ax                      ; Save the AX register value
 0001  53                                push    bx                      ; Save the BX register value
 0002  51                                push    cx                      ; Save the CX register value
 0003  52                                push    dx                      ; Save the DX register value
 0004  1E                                push    ds                      ; Save the DS register value
 0005  B8 0040                           mov     ax,BIOS_SEG             ; Get the BIOS data area segment value
 0008  8E D8                             mov     ds,ax                   ; Set DS to the BIOS data area segment
 000A  B4 0F                             mov     ah,GET_MODE             ; Setup to get current video mode
 000C  CD 10                             int     VIDEO                   ; Get the current video mode
 000E  24 7F                             and     al,MODE_MASK            ; Mask off the clear regen bit (EGA)
 0010  8A D4                             mov     dl,ah                   ; Get number of columns on the screen
```

```
Microsoft (R) Macro Assembler  Version 4.00             9/9/86 12:00:00

Clear_Screen    Sample Clear Screen Routine             Page     1-4


 0012  FE CA                      dec    dl               ; Decrement to get columns - 1
 0014  8A 36 0084                 mov    dh,ds:[V_ROWS]   ; Set number of rows on the screen
 0018  0A F6                      or     dh,dh            ; Check to see if number of rows zero
 001A  75 02                      jnz    Get_Attr         ; Jump if a non-zero number of rows
 001C  B6 18                      mov    dh,DEFAULT_ROWS  ; Set number of rows to default (24)
 001E                    Get_Attr:
 001E  B7 07                      mov    bh,TEXT_ATTR     ; Default to the text mode attribute
 0020  3C 03                      cmp    al,TEXT_MODE     ; Check current mode for text mode
 0022  76 06                      jbe    Do_Setup         ; Jump if current mode is a text mode
 0024  3C 07                      cmp    al,MONO_MODE     ; Check for monochrome text mode
 0026  74 02                      je     Do_Setup         ; Jump if current mode is monochrome
 0028  B7 00                      mov    bh,GRAPHICS_ATTR ; Graphics mode, use the graphics attr.
 002A                    Do_Setup:
 002A  B4 06                      mov    ah,SCROLL_UP     ; Setup for a BIOS scroll up call
 002C  32 C0                      xor    al,al            ; Setup to scroll the entire window
 002E  33 C9                      xor    cx,cx            ; Set upper left to 0,0
 0030  CD 10                      int    VIDEO            ; Do the BIOS video scroll up call
 0032  1F                         pop    ds               ; Restore the DS register value
 0033  5A                         pop    dx               ; Restore the DX register value
 0034  59                         pop    cx               ; Restore the CX register value
 0035  5B                         pop    bx               ; Restore the BX register value
 0036  58                         pop    ax               ; Restore the AX register value
 0037  C3                         ret                     ; Return to the caller
                         Clear_Screen    Endp             ; End of the clear screen procedure
                         ;
                         ;  Define the end of the code segment
```

```
Microsoft (R) Macro Assembler  Version 4.00              9/9/86 12:00:00

Clear_Screen    Sample Clear Screen Routine              Page     1-5


                              ;
 0038                         code    ends
                                      end                          ; End of the clear screen module
```

B.2 LOADING THE CHARACTER FONT

Characters displayed in the text modes have their font stored in plane 2 of the video RAM. Unlike previous video controllers that used a ROM as the character generator, this arrangement allows software to load special fonts, such as icons used in customized applications and national character sets. The BIOS, has the flexibility to allow selective replacement of some or all of the working character fonts.

These examples show how the data structures necessary for supporting a user-supplied font are loaded with BIOS calls. Both 25- and 43-line text modes are illustrated.

```
Microsoft (R) Macro Assembler  Version 4.00                9/9/86 12:00:00

Load_Font       Sample Load Font Routines                  Page     1-1


                                    page    30,120
                                    title   Load_Font       Sample Load Font Routines
                              ;*****************************************************************************
                              ;
                              ;  Module Functional Description:
                              ;
                              ;               This module contains routines to load character fonts
                              ;       on the COMPAQ Enhanced Color Graphics (CECG) Board. This module
                              ;       contains the following routines:
                              ;
                              ;               1) Load_8x8   - Loads the ROM 8x8 double dot font
                              ;
                              ;               2) Load_8x14  - Loads the ROM 8x14 font
                              ;
                              ;               3) Patch_Char - Patches a character in the current font
                              ;
                              ;*****************************************************************************
                                    public  Load_8x8                ; Load ROM 8x8 font routine
                                    public  Load_8x14               ; Load ROM 8x14 font routine
                                    public  Patch_Char              ; Patch a font character routine
                                    extrn   Present:near            ; Check enhanced presence routine
                                    extrn   Active:near             ; Check enhanced active routine
                              ;
                              ;  LOCAL Equates
                              ;
 = 0003                       TEXT_MODE       equ     03h           ; Maximum color/graphics text mode
```

```
Microsoft (R) Macro Assembler  Version 4.00              9/9/86 12:00:00

Load_Font       Sample Load Font Routines                Page     1-2


 = 0007                        MONO_MODE       equ     07h             ; Monochrome text mode
 = 0010                        VIDEO           equ     10h             ; BIOS video interrupt number
 = 0002                        SET_CURSOR      equ     02h             ; Set cursor position function code
 = 0003                        GET_CURSOR      equ     03h             ; Get cursor position function code
 = 000F                        GET_MODE        equ     0Fh             ; Get current video mode function code
 = 0011                        LOAD_FONT       equ     11h             ; Load font function code
 = 0000                        USER_FONT       equ     00h             ; Load user font function code
 = 0010                        RE_CALC         equ     10h             ; Recalculate video parameters bit
 = 0030                        RET_FONT_INFO   equ     30h             ; Return font information sub-function
 = 007F                        MODE_MASK       equ     7Fh             ; Current video mode mask value
 = 0000                        RET_1F          equ     0h              ; Return INT 1Fh pointer
 = 0001                        RET_43          equ     1h              ; Return INT 43h pointer
 = 0002                        RET_8x14        equ     2h              ; Return ROM 8x14 font pointer
 = 0003                        RET_8x8_All     equ     3h              ; Return ROM 8x8 pointer (Complete)
 = 0004                        RET_8x8_Up      equ     4h              ; Return ROM 8x8 pointer (Upper 128)
 = 0005                        RET_9x14        equ     5h              ; Return ROM 9x14 patch font
 = 0000                        DEFAULT_BLOCK   equ     0h              ; Default block specifier (Block 0)
 = 0100                        DEFAULT_COUNT   equ     100h            ; Default character count (256)
 = 0000                        DEFAULT_OFFSET  equ     0h              ; Default character offset (0)
                               ;
                               ;  Define the code segment
                               ;
 0000                          code    segment byte public 'CODE'      ; Standard code segment
                                       assume  cs:code, ds:nothing
```

```
Microsoft (R) Macro Assembler  Version 4.00              9/9/86 12:00:00

Load_Font       Sample Load Font Routines                Page     1-3


                                          page
                                  ;***************************************************************************
                                  ;
                                  ;       Load_8x8()                      The following table shows the number
                                  ;                                       of rows on the screen for the two
                                  ;       Registers on Entry:             different scan rates.
                                  ;
                                  ;               None                    Scan Rate               Rows/Screen
                                  ;                                       ---------               -----------
                                  ;       Registers on Exit:
                                  ;                                         200 Scan lines        25 Rows/Screen
                                  ;               None                      350 Scan lines        43 Rows/Screen
                                  ;
                                  ;***************************************************************************
 0000                             Load_8x8        Proc    Near            ; Load ROM 8x8 font procedure
 0000  50                                 push    ax                      ; Save the AX register value
 0001  53                                 push    bx                      ; Save the BX register value
 0002  51                                 push    cx                      ; Save the CX register value
 0003  52                                 push    dx                      ; Save the DX register value
 0004  55                                 push    bp                      ; Save the BP register value
 0005  06                                 push    es                      ; Save the ES register value
 0006  E8 0000 E                          call    Present                 ; Check for enhanced adapter present
 0009  72 37                              jc      Return_8x8              ; Jump if NO enhanced adapter
 000B  E8 0000 E                          call    Active                  ; Check for enhanced adapter active
 000E  72 32                              jc      Return_8x8              ; Jump if NOT the active adapter
 0010  B4 11                              mov     ah,LOAD_FONT            ; Get the load font function code
```

```
Microsoft (R) Macro Assembler  Version 4.00              9/9/86 12:00:00

Load_Font       Sample Load Font Routines                 Page     1-4


 0012  B0 30                         mov     al,RET_FONT_INFO        ; Get return font info. sub-function
 0014  B7 03                         mov     bh,RET_8x8_All          ; Setup to get pointer to ROM 8x8 font
 0016  CD 10                         int     VIDEO                   ; Get ROM 8x8 font information
 0018  B4 11                         mov     ah,LOAD_FONT            ; Get the load font function code
 001A  B0 10                         mov     al,USER_FONT+RE_CALC    ; Get load user font sub-function code
 001C  B7 08                         mov     bh,8                    ; Move 8 to bytes per character
 001E  B3 00                         mov     bl,DEFAULT_BLOCK        ; Set the default block specifier (0)
 0020  B9 0100                       mov     cx,DEFAULT_COUNT        ; Set the default character count (256)
 0023  BA 0000                       mov     dx,DEFAULT_OFFSET       ; Set the default character offset (0)
 0026  CD 10                         int     VIDEO                   ; Load the user (8x8) character font
 0028  B4 11                         mov     ah,LOAD_FONT            ; Get the load font function code
 002A  B0 30                         mov     al,RET_FONT_INFO        ; Get return font info. sub-function
 002C  B7 01                         mov     bh,RET_43               ; Setup to get INT 43 information
 002E  CD 10                         int     VIDEO                   ; Get the INT 43 video information
 0030  8A DA                         mov     bl,dl                   ; Save the number of rows on the screen
 0032  B4 03                         mov     ah,GET_CURSOR           ; Get the get cursor position function
 0034  32 FF                         xor     bh,bh                   ; Setup to use the current video page
 0036  CD 10                         int     VIDEO                   ; Get the current cursor position
 0038  3A F3                         cmp     dh,bl                   ; Check the current cursor row value
 003A  72 06                         jb      Return_8x8              ; Jump if below the maximum value
 003C  8A F3                         mov     dh,bl                   ; Setup row value to maximum possible
 003E  B4 02                         mov     ah,SET_CURSOR           ; Get set cursor position function
 0040  CD 10                         int     VIDEO                   ; Set cursor position to last row
 0042                        Return_8x8:
 0042  07                            pop     es                      ; Restore the ES register value
 0043  5D                            pop     bp                      ; Restore the BP register value
```

```
Microsoft (R) Macro Assembler  Version 4.00                9/9/86 12:00:00

Load_Font       Sample Load Font Routines                   Page     1-5


 0044  5A                            pop     dx                   ; Restore the DX register value
 0045  59                            pop     cx                   ; Restore the CX register value
 0046  5B                            pop     bx                   ; Restore the BX register value
 0047  58                            pop     ax                   ; Restore the AX register value
 0048  C3                            ret                          ; Return to the caller
                             Load_8x8        Endp                 ; End of the load ROM 8x8 procedure
```

```
Microsoft (R) Macro Assembler  Version 4.00              9/9/86 12:00:00

Load_Font       Sample Load Font Routines                Page     1-6


                                page
                              ;*****************************************************************************
                              ;
                              ;       Load_8x14()                     The following table shows the number
                              ;                                       of rows on the screen for the two
                              ;       Registers on Entry:             different scan rates.
                              ;
                              ;               None                    Scan Rate               Rows/Screen
                              ;                                       ---------               -----------
                              ;       Registers on Exit:
                              ;                                         200 Scan lines        14 Rows/Screen
                              ;               None                      350 Scan lines        25 Rows/Screen
                              ;
                              ;       Note - This routine will load the 9x14 font (Automatically via the ROM)
                              ;              if the current mode is monochrome text.
                              ;
                              ;*****************************************************************************
 0049                         Load_8x14       Proc    Near            ; Load ROM 8x14 font procedure
 0049  50                             push    ax                      ; Save the AX register value
 004A  53                             push    bx                      ; Save the BX register value
 004B  51                             push    cx                      ; Save the CX register value
 004C  52                             push    dx                      ; Save the DX register value
 004D  55                             push    bp                      ; Save the BP register value
 004E  06                             push    es                      ; Save the ES register value
 004F  E8 0000 E                      call    Present                 ; Check for enhanced adapter present
 0052  72 37                          jc      Return_8x14             ; Jump if NO enhanced adapter
```

LOADING THE CHARACTER FONT

```
Microsoft (R) Macro Assembler  Version 4.00              9/9/86 12:00:00

Load_Font        Sample Load Font Routines               Page     1-7


 0054  E8 0000 E                     call    Active                   ; Check for enhanced adapter active
 0057  72 32                         jc      Return_8x14              ; Jump if NOT the active adapter
 0059  B4 11                         mov     ah,LOAD_FONT             ; Get the load font function code
 005B  B0 30                         mov     al,RET_FONT_INFO         ; Get return font info. sub-function
 005D  B7 02                         mov     bh,RET_8x14              ; Setup to get pointer to ROM 8x14 font
 005F  CD 10                         int     VIDEO                    ; Get ROM 8x14 font information
 0061  B4 11                         mov     ah,LOAD_FONT             ; Get the load font function code
 0063  B0 10                         mov     al,USER_FONT+RE_CALC     ; Get load user font sub-function code
 0065  B7 0E                         mov     bh,14                    ; Move 14 to bytes per character
 0067  B3 00                         mov     bl,DEFAULT_BLOCK         ; Set the default block specifier (0)
 0069  B9 0100                       mov     cx,DEFAULT_COUNT         ; Set the default character count (256)
 006C  BA 0000                       mov     dx,DEFAULT_OFFSET        ; Set the default character offset (0)
 006F  CD 10                         int     VIDEO                    ; Load the user (8x14) character font
 0071  B4 11                         mov     ah,LOAD_FONT             ; Get the load font function code
 0073  B0 30                         mov     al,RET_FONT_INFO         ; Get return font info. sub-function
 0075  B7 01                         mov     bh,RET_43                ; Setup to get INT 43 information
 0077  CD 10                         int     VIDEO                    ; Get the INT 43 video information
 0079  8A DA                         mov     bl,dl                    ; Save the number of rows on the screen
 007B  B4 03                         mov     ah,GET_CURSOR            ; Get the get cursor position function
 007D  32 FF                         xor     bh,bh                    ; Setup to use the current video page
 007F  CD 10                         int     VIDEO                    ; Get the current cursor position
 0081  3A F3                         cmp     dh,bl                    ; Check the current cursor row value
 0083  72 06                         jb      Return_8x14              ; Jump if below the maximum value
 0085  8A F3                         mov     dh,bl                    ; Setup row value to maximum possible
 0087  B4 02                         mov     ah,SET_CURSOR            ; Get set cursor position function
 0089  CD 10                         int     VIDEO                    ; Set cursor position to last row
```

```
Microsoft (R) Macro Assembler  Version 4.00               9/9/86 12:00:00

Load_Font       Sample Load Font Routines                 Page     1-8


 008B                         Return_8x14:
 008B  07                             pop     es                       ; Restore the ES register value
 008C  5D                             pop     bp                       ; Restore the BP register value
 008D  5A                             pop     dx                       ; Restore the DX register value
 008E  59                             pop     cx                       ; Restore the CX register value
 008F  5B                             pop     bx                       ; Restore the BX register value
 0090  58                             pop     ax                       ; Restore the AX register value
 0091  C3                             ret                              ; Return to the caller
                              Load_8x14       Endp                     ; End of the load ROM 8x14 procedure
```

LOADING THE CHARACTER FONT

```
Microsoft (R) Macro Assembler  Version 4.00              9/9/86 12:00:00

Load_Font       Sample Load Font Routines                Page     1-9


                                    page
                              ;*****************************************************************************
                              ;
                              ;     Patch_Char(Character_Code, Character_Pattern)   This call will have
                              ;                                                      NO effect if in a
                              ;     Registers on Entry:                              graphics video mode.
                              ;
                              ;            AL    - Character code to patch
                              ;            ES:BP - Pointer to character dot pattern
                              ;
                              ;     Registers on Exit:
                              ;
                              ;            None
                              ;
                              ;*****************************************************************************
 0092                         Patch_Char     Proc    Near             ; Patch font character procedure
 0092  50                           push     ax                       ; Save the AX register value
 0093  53                           push     bx                       ; Save the BX register value
 0094  51                           push     cx                       ; Save the CX register value
 0095  52                           push     dx                       ; Save the DX register value
 0096  E8 0000 E                    call     Present                  ; Check for enhanced adapter present
 0099  72 32                        jc       No_Patch                 ; Jump if NO enhanced adapter
 009B  E8 0000 E                    call     Active                   ; Check for enhanced adapter active
 009E  72 2D                        jc       No_Patch                 ; Jump if NOT the active adapter
 00A0  8A F0                        mov      dh,al                    ; Set DH to the character code to patch
 00A2  B4 0F                        mov      ah,GET_MODE              ; Get the get mode function code
```

```
Microsoft (R) Macro Assembler  Version 4.00             9/9/86 12:00:00

Load_Font       Sample Load Font Routines               Page     1-10


 00A4  CD 10                     int     VIDEO                ; Get the current video mode
 00A6  24 7F                     and     al,MODE_MASK         ; Strip off the clear regen flag
 00A8  3C 07                     cmp     al,MONO_MODE         ; Check for monochrome text mode
 00AA  74 04                     je      Do_Patch             ; Jump if in monochrome text mode
 00AC  3C 03                     cmp     al,TEXT_MODE         ; Check for color/graphics text mode
 00AE  77 1D                     ja      No_Patch             ; Jump if NOT in a text video mode
 00B0                   Do_Patch:
 00B0  06                        push    es                   ; Save the character patch segment
 00B1  55                        push    bp                   ; Save the character patch offset
 00B2  B4 11                     mov     ah,LOAD_FONT         ; Get the load font function code
 00B4  B0 30                     mov     al,RET_FONT_INFO     ; Get return font info. sub-function
 00B6  B7 01                     mov     bh,RET_43            ; Setup to get INT 43 information
 00B8  CD 10                     int     VIDEO                ; Get the INT 43 video information

 00BA  5D                        pop     bp                   ; Restore the character patch offset
 00BB  07                        pop     es                   ; Restore the character patch segment
 00BC  8A F9                     mov     bh,cl                ; Move bytes per character to BH
 00BE  32 DB                     xor     bl,bl                ; Setup to modify character in block 0
 00C0  B9 0001                   mov     cx,1                 ; Setup to modify 1 character pattern
 00C3  8A D6                     mov     dl,dh                ; Move character to patch into DL
 00C5  32 F6                     xor     dh,dh                ; Setup the character offset in DX
 00C7  B4 11                     mov     ah,LOAD_FONT         ; Get the load font function code
 00C9  B0 00                     mov     al,USER_FONT         ; Get load user font sub-function code
 00CB  CD 10                     int     VIDEO                ; Modify the desired character pattern
 00CD                   No_Patch:
 00CD  5A                        pop     dx                   ; Restore the DX register value
 00CE  59                        pop     cx                   ; Restore the CX register value
```

```
Microsoft (R) Macro Assembler  Version 4.00            9/9/86 12:00:00

Load_Font       Sample Load Font Routines              Page     1-11


 00CF  5B                          pop     bx                   ; Restore the BX register value
 00D0  58                          pop     ax                   ; Restore the AX register value
 00D1  C3                          ret                          ; Return to the caller
                           Patch_Char      Endp                 ; End of the patch character procedure
                           ;
                           ;  Define the end of the code segment
                           ;
 00D2                      code    ends
                                   end                          ; End of the load font module
```

B.3 SHOWING A SPLIT SCREEN

The split-screen feature is supported by hardware that allows an arbitrarily positioned boundary between upper and lower portions of the screen. Each portion displays from a different block of video RAM. Thus, one or two software processes can update the video memory independently without regard to where display-side software has split the screen. The feature can be used in any of the text or graphics formats.

As shown in Figure B-1, Screen B shows data beginning at an address that can be the start of a logical window. (See section B.4.) This address is typically nonzero.

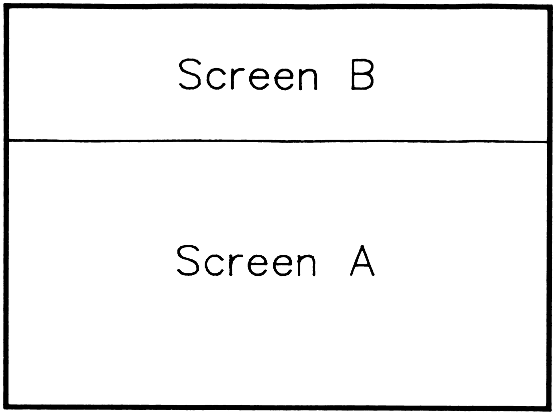


Figure B-1. Example of a Split Screen

The data shown in Screen A always begins in video memory at offset 0000h, irrespective of the physical positioning of the split.

SHOWING A SPLIT SCREEN

```
Microsoft (R) Macro Assembler  Version 4.00                9/9/86 12:00:00

Split_Screen    Sample Split Screen Routine                Page     1-1


                                        page    30,120
                                        title   Split_Screen    Sample Split Screen Routine
                                ;*****************************************************************************
                                ;
                                ;  Module Functional Description:
                                ;
                                ;               This module contains a routine to perform vertical split
                                ;       screens on the COMPAQ Enhanced Color Graphics (CECG) Board. This
                                ;       routine functions in both the text and graphics modes.
                                ;
                                ;*****************************************************************************
                                        public  Split_Screen            ; Split screen routine
                                        extrn   Present:near            ; Check enhanced presence routine
                                        extrn   Active:near             ; Check enhanced active routine
                                        extrn   Actual:near             ; Get actual mode value routine
                                        extrn   Out_Pair:near           ; Output register pair routine
                                ;
                                ;  LOCAL Equates
                                ;
 = 0040                         BIOS_SEG        equ     40h             ; ROM BIOS data area segment
 = 0010                         VIDEO           equ     10h             ; BIOS video interrupt number
 = 000F                         GET_MODE        equ     0Fh             ; Get current video mode function code
 = 0011                         LOAD_FONT       equ     11h             ; Load font function code
 = 0030                         RET_FONT_INFO   equ     30h             ; Return font information sub-function
 = 007F                         MODE_MASK       equ     7Fh             ; Current video mode mask value
 = 0003                         TEXT_MODE       equ     03h             ; Maximum color text video mode value
```

```
Microsoft (R) Macro Assembler  Version 4.00               9/9/86 12:00:00

Split_Screen    Sample Split Screen Routine               Page     1-2


 = 0007                     MONO_MODE       equ     07h             ; Monochrome text video mode value
 = 0000                     V_PARM          equ     00h             ; Offset video parm. ptr. in save table
 = 01FF                     MAX_ROW         equ     1FFh            ; Maximum scan line row value
 = 0000                     RET_1F          equ     00h             ; Return INT 1Fh pointer
 = 0001                     RET_43          equ     01h             ; Return INT 43h pointer
 = 0002                     RET_8x14        equ     02h             ; Return ROM 8x14 font pointer
 = 0003                     RET_8x8_All     equ     03h             ; Return ROM 8x8 pointer (Complete)
 = 0004                     RET_8x8_Up      equ     04h             ; Return ROM 8x8 pointer (Upper 128)
 = 0005                     RET_9x14        equ     05h             ; Return ROM 9x14 patch font
 = 000A                     CRT_PARM        equ     0Ah             ; CRT controller parameters offset
 = 0007                     OVER_FLOW       equ     07h             ; CRT controller overflow register
 = 0018                     LINE_COMPARE    equ     18h             ; CRT controller line compare register
 = 0011                     OVERFLOW        equ     CRT_PARM+OVER_FLOW
 = 00EF                     LINE_MASK       equ     0EFh            ; Line compare bit mask value
 = 0004                     LINE_SHIFT      equ     4h              ; Line compare bit shift value
 = 0063                     v_base          equ     word ptr 063h   ; CRT controller base address offset
 = 00A8                     save_ptr        equ     dword ptr 0A8h  ; Save area table offset in BIOS data
                            ;
                            ;  Define the code segment
                            ;
 0000                       code    segment byte public 'CODE'      ; Standard code segment
                                    assume  cs:code, ds:nothing
```

SHOWING A SPLIT SCREEN

```
Microsoft (R) Macro Assembler  Version 4.00                 9/9/86 12:00:00

Split_Screen    Sample Split Screen Routine                 Page     1-3


                                        page
                                ;***************************************************************************
                                ;
                                ;       Split_Screen(Row)
                                ;
                                ;       Registers on Entry:
                                ;
                                ;               AX    - Row number      (Character row for text,
                                ;                       (0=No split)     scan line for graphics)
                                ;
                                ;       Registers on Exit:
                                ;
                                ;               None
                                ;
                                ;***************************************************************************
 0000                           Split_Screen    Proc    Near            ; Split screen procedure
 0000  50                               push    ax                      ; Save the AX register value
 0001  53                               push    bx                      ; Save the BX register value
 0002  51                               push    cx                      ; Save the CX register value
 0003  52                               push    dx                      ; Save the DX register value
 0004  56                               push    si                      ; Save the SI register value
 0005  55                               push    bp                      ; Save the BP register value
 0006  1E                               push    ds                      ; Save the DS register value
 0007  06                               push    es                      ; Save the ES register value
 0008  E8 0000 E                        call    Present                 ; Check for enhanced adapter present
 000B  72 57                            jc      Return_Split            ; Jump if NO enhanced adapter present
```

```
Microsoft (R) Macro Assembler  Version 4.00                9/9/86 12:00:00

Split_Screen    Sample Split Screen Routine                Page     1-4


 000D  E8 0000 E                call    Active                  ; Check for enhanced adapter active
 0010  72 52                    jc      Return_Split            ; Jump if enhanced adapter NOT active
 0012  BB 0040                  mov     bx,BIOS_SEG             ; Get the BIOS data area segment
 0015  8E DB                    mov     ds,bx                   ; Setup to access the BIOS data area
 0017  8B C8                    mov     cx,ax                   ; Save the row number in CX
 0019  B4 0F                    mov     ah,GET_MODE             ; Get the BIOS get video mode function
 001B  CD 10                    int     VIDEO                   ; Get the current video mode value
 001D  24 7F                    and     al,MODE_MASK            ; Mask off all but the video mode
 001F  3C 03                    cmp     al,TEXT_MODE            ; Check for a color/graphics text mode
 0021  76 4A                    jbe     Text_Split              ; Jump if a color/graphics text mode
 0023  3C 07                    cmp     al,MONO_MODE            ; Check for a monochrome text mode
 0025  74 46                    je      Text_Split              ; Jump if monochrome text mode
 0027                   Graphics_Split:
 0027  0B C9                    or      cx,cx                   ; Check for NO split value
 0029  75 03                    jnz     Do_Split                ; Jump if NOT the no split value
 002B  B9 01FF                  mov     cx,MAX_ROW              ; No split desired, get max row value
 002E                   Do_Split:
 002E  81 E1 01FF               and     cx,MAX_ROW              ; Mask off all but the row bits
 0032  49                       dec     cx                      ; Compute the actual output value
 0033  E8 0000 E                call    Actual                  ; Call routine to get actual mode value
 0036  C4 1E 00A8               les     bx,ds:[save_ptr]        ; Get the save area table pointer
 003A  26: C4 1F                les     bx,es:[bx+V_PARM]       ; Get the video parameter table pointer
 003D  98                       cbw                             ; Convert actual mode to full word
 003E  86 C4                    xchg    al,ah                   ; Get the actual mode * 256
 0040  D1 E8                    shr     ax,1                    ; Get actual mode * 128
 0042  D1 E8                    shr     ax,1                    ; Get mode * 64 (Sub-table size)
```

```
Microsoft (R) Macro Assembler  Version 4.00                 9/9/86 12:00:00

Split_Screen    Sample Split Screen Routine                 Page     1-5


 0044  8B F0                    mov     si,ax                   ; Move sub-table offset into SI
 0046  26: 8A 40 11             mov     al,es:[bx+si+OVERFLOW]  ; Get the overflow value for this mode
 004A  8B D9                    mov     bx,cx                   ; Save the row value in BX
 004C  24 EF                    and     al,LINE_MASK            ; Mask off the current line compare bit
 004E  B1 04                    mov     cl,LINE_SHIFT           ; Get the line compare bit shift value
 0050  D2 E7                    shl     bh,cl                   ; Shift line compare bit into position
 0052  0A C7                    or      al,bh                   ; Or in the new line compare bit
 0054  8B 16 0063               mov     dx,ds:[v_base]          ; Get the CRT controller base I/O addr.
 0058  B4 07                    mov     ah,OVER_FLOW            ; Get the overflow register index
 005A  E8 0000 E                call    Out_Pair                ; Call routine to output register pair
 005D  B4 18                    mov     ah,LINE_COMPARE         ; Get the line compare register index
 005F  8A C3                    mov     al,bl                   ; Get the new line compare value
 0061  E8 0000 E                call    Out_Pair                ; Call routine to output register pair
 0064                   Return_Split:
 0064  07                       pop     es                      ; Restore the ES register value
 0065  1F                       pop     ds                      ; Restore the DS register value
 0066  5D                       pop     bp                      ; Restore the BP register value
 0067  5E                       pop     si                      ; Restore the SI register value
 0068  5A                       pop     dx                      ; Restore the DX register value
 0069  59                       pop     cx                      ; Restore the CX register value
 006A  5B                       pop     bx                      ; Restore the BX register value
 006B  58                       pop     ax                      ; Restore the AX register value
 006C  C3                       ret                             ; Return to the caller
 006D                   Text_Split:
 006D  50                       push    ax                      ; Save the current video mode value
 006E  51                       push    cx                      ; Save the desired row split number
```

```
Microsoft (R) Macro Assembler  Version 4.00              9/9/86 12:00:00

Split_Screen    Sample Split Screen Routine              Page     1-6


 006F  B4 11                  mov    ah,LOAD_FONT         ; Get the load font function code
 0071  B0 30                  mov    al,RET_FONT_INFO     ; Get return font info. sub-function
 0073  B7 01                  mov    bh,RET_43            ; Setup to get INT 43 information
 0075  CD 10                  int    VIDEO                ; Get the INT 43 video information
 0077  58                     pop    ax                   ; Restore the desired row split number
 0078  5B                     pop    bx                   ; Restore the current video mode value
 0079  3A C2                  cmp    al,dl                ; Check against maximum row number
 007B  76 03                  jbe    Get_Scan_Line        ; Jump if above the maximum row
 007D  F9                     stc                         ; Set carry flag indicating error
 007E  EB E4                  jmp    Return_Split         ; Go return to the caller
 0080                 Get_Scan_Line:
 0080  F7 E1                  mul    cx                   ; Multiply by points to get scan line
 0082  0B C0                  or     ax,ax                ; Check for no split desired value
 0084  75 03                  jnz    Save_Row             ; Jump if a split is desired
 0086  B8 01FF                mov    ax,MAX_ROW           ; No split desired, get max row value
 0089                 Save_Row:
 0089  8B C8                  mov    cx,ax                ; Move desired row number to CX
 008B  8B C3                  mov    ax,bx                ; Move current video mode into AX
 008D  EB 9F                  jmp    SHORT Do_Split       ; Go do the desired split
                      Split_Screen   Endp                 ; End of the split screen procedure
                      ;
                      ;  Define the end of the code segment
                      ;
 008F                 code    ends
                              end                         ; End of the split screen module
```

B.4 LOGICAL WINDOWING

Logical windowing refers to the process of selectively displaying a portion of an already-formatted larger area from video memory. This means that video memory can be organized to have more columns or rows than are visible. By simply changing register contents, a different portion of video memory can be displayed. This saves the time-consuming process of having to move the data for a fixed window.

This routine sets up a logical line length greater than the physical line length, and is based on the current video mode.

Figure B-2 shows the relationship between a window and its borders.

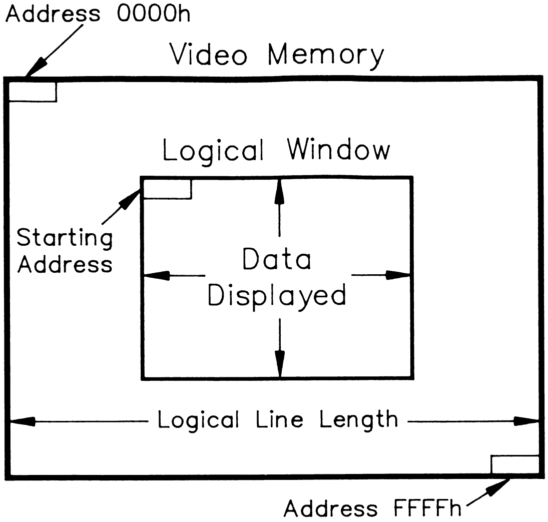

Address 0000h
Video Memory
Logical Window
Starting Address
Data Displayed
Logical Line Length
Address FFFFh

Figure B-2. Relationship Between a Logical Window and Its Borders

```
Microsoft (R) Macro Assembler  Version 4.00                 9/9/86 12:00:00

Logical_Window  Sample Logical Windowing Routine            Page     1-1


                                        page    30,120
                                        title   Logical_Window  Sample Logical Windowing Routine
                                ;*******************************************************************************
                                ;
                                ;  Module Functional Description:
                                ;
                                ;               This module contains a routine to perform logical windowing
                                ;       on the COMPAQ Enhanced Color Graphics (CECG) Board. This routine
                                ;       functions in both the text and graphics modes and sets the logical
                                ;       window size to the maximum amount of video RAM available.
                                ;
                                ;*******************************************************************************
                                        public  Logical_Window          ; Logical windowing routine
                                        extrn   Present:near            ; Check enhanced presence routine
                                        extrn   Active:near             ; Check enhanced active routine
                                        extrn   Actual:near             ; Get actual mode value routine
                                        extrn   Out_Pair:near           ; Output register pair routine
                                ;
                                ;  LOCAL Equates
                                ;
 = 0040                         BIOS_SEG        equ     40h             ; ROM BIOS data area segment
 = 0010                         VIDEO           equ     10h             ; BIOS video interrupt number
 = 000F                         GET_MODE        equ     0Fh             ; Get current video mode function code
 = 007F                         MODE_MASK       equ     7Fh             ; Current video mode mask value
 = 0013                         OFF_SET         equ     13h             ; CRT controller offset register
 = 0063                         v_base          equ     word ptr 063h   ; CRT controller base address offset
```

```
Microsoft (R) Macro Assembler  Version 4.00               9/9/86 12:00:00

Logical_Window  Sample Logical Windowing Routine          Page     1-2


                                  ;
                                  ;  Define the code segment
                                  ;
 0000                             code    segment byte public 'CODE'      ; Standard code segment
                                          assume  cs:code, ds:nothing
```

```
Microsoft (R) Macro Assembler  Version 4.00            9/9/86 12:00:00

Logical_Window  Sample Logical Windowing Routine       Page     1-3


                                  page
                                 ;****************************************************************************
                                 ;
                                 ;       Logical_Window()
                                 ;
                                 ;       Registers on Entry:
                                 ;
                                 ;               None
                                 ;
                                 ;       Registers on Exit:
                                 ;
                                 ;               AH    - Number of screens horizontally
                                 ;               AL    - Number of screens vertically
                                 ;               CX    - Logical line size (Bytes)
                                 ;
                                 ;****************************************************************************
 0000                            Logical_Window  Proc    Near            ; Logical windowing procedure
 0000  53                                push    bx                      ; Save the BX register value
 0001  52                                push    dx                      ; Save the DX register value
 0002  1E                                push    ds                      ; Save the DS register value
 0003  E8 0000 E                         call    Present                 ; Check for enhanced adapter present
 0006  72 30                             jc      Return_Logical          ; Jump if NO enhanced adapter present
 0008  E8 0000 E                         call    Active                  ; Check for enhanced adapter active
 000B  72 2B                             jc      Return_Logical          ; Jump if enhanced adapter NOT active
 000D  B4 0F                             mov     ah,GET_MODE             ; Get the BIOS get video mode function
 000F  CD 10                             int     VIDEO                   ; Get the current video mode value
```

```
Microsoft (R) Macro Assembler  Version 4.00             9/9/86 12:00:00

Logical_Window  Sample Logical Windowing Routine        Page     1-4


 0011  24 7F                       and     al,MODE_MASK           ; Mask off all but the video mode
 0013  E8 0000 E                   call    actual                 ; Call routine to get actual mode
 0016  98                          cbw                            ; Convert video mode to full word
 0017  8B D8                       mov     bx,ax                  ; Move the video mode value to BX
 0019  B8 0040                     mov     ax,BIOS_SEG            ; Get the BIOS data segment value
 001C  8E D8                       mov     ds,ax                  ; Setup to access the BIOS data area
 001E  8B 16 0063                  mov     dx,ds:[v_base]         ; Get the CRTC base port address
 0022  2E: 8A 87 003C R            mov     al,cs:[Length_Table+bx] ; Get the new logical line length
 0027  B4 13                       mov     ah,OFF_SET             ; Setup to output the offset register
 0029  E8 0000 E                   call    Out_Pair               ; Output the new logical line width
 002C  D1 E3                       shl     bx,1                   ; Convert to word table index value
 002E  2E: 8B 87 0053 R            mov     ax,cs:[Screen_Table+bx] ; Get the number of screens available
 0033  2E: 8B 8F 0081 R            mov     cx,cs:[Logical_table+bx]; Get new logical line size (Bytes)
 0038                      Return_Logical:
 0038  1F                          pop     ds                     ; Restore the DS register value
 0039  5A                          pop     dx                     ; Restore the DX register value
 003A  5B                          pop     bx                     ; Restore the BX register value
 003B  C3                          ret                            ; Return to the caller
                           Logical_Window  Endp                   ; End of the logical window procedure
                           ;
                           ;  Define the logical line length table (All possible modes)
                           ;
 = 003C                    Length_Table    Equ     This Byte      ; Logical line length table
 003C  50                          db      050h                   ; Video mode 0  logical line length
 003D  50                          db      050h                   ; Video mode 1  logical line length
 003E  A0                          db      0A0h                   ; Video mode 2  logical line length
```

```
Microsoft (R) Macro Assembler  Version 4.00              9/9/86 12:00:00

Logical_Window  Sample Logical Windowing Routine         Page     1-5


 003F  A0                          db      0A0h              ; Video mode 3   logical line length
 0040  14                          db      014h              ; Video mode 4   logical line length
 0041  14                          db      014h              ; Video mode 5   logical line length
 0042  28                          db      028h              ; Video mode 6   logical line length
 0043  A0                          db      0A0h              ; Video mode 7   logical line length
 0044  50                          db      050h              ; Video mode 8   logical line length
 0045  50                          db      050h              ; Video mode 9   logical line length
 0046  50                          db      050h              ; Video mode A   logical line length
 0047  A0                          db      0A0h              ; Video mode B   logical line length
 0048  A0                          db      0A0h              ; Video mode C   logical line length
 0049  50                          db      050h              ; Video mode D   logical line length
 004A  50                          db      050h              ; Video mode E   logical line length
 004B  28                          db      028h              ; Video mode F   logical line length
 004C  28                          db      028h              ; Video mode 10  logical line length
 004D  50                          db      050h              ; Video mode F+  logical line length
 004E  50                          db      050h              ; Video mode 10+ logical line length
 004F  50                          db      050h              ; Video mode 0*  logical line length
 0050  50                          db      050h              ; Video mode 1*  logical line length
 0051  A0                          db      0A0h              ; Video mode 2*  logical line length
 0052  A0                          db      0A0h              ; Video mode 3*  logical line length
                        ;
                        ;  Define the logical screen size table (All possible modes)
                        ;
 = 0053                 Screen_Table    Equ     This Word    ; Logical screen size table
 0053  04 04                       db      4,4               ; Mode 0   screen size (Vert,Horiz)
 0055  04 04                       db      4,4               ; Mode 1   screen size (Vert,Horiz)
```

```
Microsoft (R) Macro Assembler  Version 4.00              9/9/86 12:00:00

Logical_Window  Sample Logical Windowing Routine         Page    1-6


 0057  02 04                      db      2,4                ; Mode 2   screen size (Vert,Horiz)
 0059  02 04                      db      2,4                ; Mode 3   screen size (Vert,Horiz)
 005B  01 01                      db      1,1                ; Mode 4   screen size (Vert,Horiz)
 005D  01 01                      db      1,1                ; Mode 5   screen size (Vert,Horiz)
 005F  01 01                      db      1,1                ; Mode 6   screen size (Vert,Horiz)
 0061  02 04                      db      2,4                ; Mode 7   screen size (Vert,Horiz)
 0063  04 04                      db      4,4                ; Mode 8   screen size (Vert,Horiz)
 0065  04 04                      db      4,4                ; Mode 9   screen size (Vert,Horiz)
 0067  04 04                      db      4,4                ; Mode A   screen size (Vert,Horiz)
 0069  02 04                      db      2,4                ; Mode B   screen size (Vert,Horiz)
 006B  02 04                      db      2,4                ; Mode C   screen size (Vert,Horiz)
 006D  02 04                      db      2,4                ; Mode D   screen size (Vert,Horiz)
 006F  02 02                      db      2,2                ; Mode E   screen size (Vert,Horiz)
 0071  01 02                      db      1,2                ; Mode F   screen size (Vert,Horiz)
 0073  01 02                      db      1,2                ; Mode 10  screen size (Vert,Horiz)
 0075  01 02                      db      1,2                ; Mode F+  screen size (Vert,Horiz)
 0077  01 02                      db      1,2                ; Mode 10+ screen size (Vert,Horiz)
 0079  04 04                      db      4,4                ; Mode 0*  screen size (Vert,Horiz)
 007B  04 04                      db      4,4                ; Mode 1*  screen size (Vert,Horiz)
 007D  02 04                      db      2,4                ; Mode 2*  screen size (Vert,Horiz)
 007F  02 04                      db      2,4                ; Mode 3*  screen size (Vert,Horiz)
                          ;
                          ;  Define the logical line size table (All possible modes)
                          ;
 = 0081                   Logical_Table   Equu    This Word    ; Logical line size table
 0081  00A0                       dw      0A0h               ; Mode 0   logical line size (Bytes)
```

```
Microsoft (R) Macro Assembler  Version 4.00              9/9/86 12:00:00

Logical_Window  Sample Logical Windowing Routine         Page      1-7


 0083  00A0                      dw      0A0h           ; Mode 1   logical line size (Bytes)
 0085  0140                      dw      140h           ; Mode 2   logical line size (Bytes)
 0087  0140                      dw      140h           ; Mode 3   logical line size (Bytes)
 0089  0050                      dw      050h           ; Mode 4   logical line size (Bytes)
 008B  0050                      dw      050h           ; Mode 5   logical line size (Bytes)
 008D  0050                      dw      050h           ; Mode 6   logical line size (Bytes)
 008F  0140                      dw      140h           ; Mode 7   logical line size (Bytes)
 0091  00A0                      dw      0A0h           ; Mode 8   logical line size (Bytes)
 0093  00A0                      dw      0A0h           ; Mode 9   logical line size (Bytes)
 0095  00A0                      dw      0A0h           ; Mode A   logical line size (Bytes)
 0097  0140                      dw      140h           ; Mode B   logical line size (Bytes)
 0099  0140                      dw      140h           ; Mode C   logical line size (Bytes)
 009B  00A0                      dw      0A0h           ; Mode D   logical line size (Bytes)
 009D  00A0                      dw      0A0h           ; Mode E   logical line size (Bytes)
 009F  00A0                      dw      0A0h           ; Mode F   logical line size (Bytes)
 00A1  00A0                      dw      0A0h           ; Mode 10  logical line size (Bytes)
 00A3  00A0                      dw      0A0h           ; Mode F+  logical line size (Bytes)
 00A5  00A0                      dw      0A0h           ; Mode 10+ logical line size (Bytes)
 00A7  00A0                      dw      0A0h           ; Mode 0*  logical line size (Bytes)
 00A9  00A0                      dw      0A0h           ; Mode 1*  logical line size (Bytes)
 00AB  0140                      dw      140h           ; Mode 2*  logical line size (Bytes)
 00AD  0140                      dw      140h           ; Mode 3*  logical line size (Bytes)
                              ;
                              ;  Define the end of the code segment
                              ;
 00AF                         code    ends
```

```
Microsoft (R) Macro Assembler  Version 4.00                9/9/86 12:00:00

Logical_Window  Sample Logical Windowing Routine           Page     1-8


                                          end                             ; End of the logical windowing module
```

B.5 SMOOTH SCROLLING AND PANNING

Smooth scrolling and panning build on the logical window feature. Instead of allowing only character (or byte) increments for the position of the data in the window, hardware allows scan line or pixel increments. Different hardware mechanisms support the scrolling and panning features.

In text modes, smooth scrolling (vertical movement) operates by displaying the first row of characters on the screen starting one or more scan lines down the character. This implies that the bottom scan lines of the bottom row of characters are also off the screen. By beginning the display of the top row of characters at a successively greater scan line within the character, the display can appear to smoothly "roll up". When the bottommost scan line of the top row of characters has been obscured, software must simultaneously change the logical window to the next row of characters and reset the scroll to the top of the character scan lines.

Smooth scrolling in graphics modes is somewhat simpler: the starting address of the logical window is changed in an increment of the length of a logical line.

Smooth panning (horizontal movement) works in both the text and graphics modes by beginning a scan line starting with a character's scan line (or graphics byte) bits <6..0>, rather than beginning with the usual bit <7>. When the last bit of the character's scan line or graphics byte has been obscured, bit <7> is again selected as the first bit to be displayed and, as with smooth scrolling, the logical window is advanced to the next byte. These sample routines perform smooth vertical scrolling and/or horizontal panning.

```
Microsoft (R) Macro Assembler  Version 4.00             9/9/86 12:00:00

Smooth_Scroll   Sample Smooth Scrolling Routine         Page     1-1


                                        page    30,120
                                        title   Smooth_Scroll   Sample Smooth Scrolling Routine
                                ;*****************************************************************************
                                ;
                                ;  Module Functional Description:
                                ;
                                ;               This module contains routines to perform smooth scrolling
                                ;       on a logical window that is greater than the physical window (screen).
                                ;
                                ;*****************************************************************************
                                        public  Scroll_Setup            ; Smooth scroll setup routine
                                        public  Smooth_Scroll           ; Smooth scrolling routine
                                        public  Do_Scroll               ; Do scroll routine
                                        public  Logical_Width           ; Logical line width
                                        public  Unit_Horiz              ; Pixels/horizontal unit
                                        public  Unit_Vert               ; Scan lines/vertical unit
                                        public  Horiz_Size              ; Pixels/horizontal step  (Physical)
                                        public  Vert_Size               ; Scan lines/vertical step (Physical)
                                        public  Current_Horiz           ; Current horizontal pixel position
                                        public  Current_Vert            ; Current vertical scan line position
                                        public  Target_Horiz            ; Target horizontal pixel position
                                        public  Target_Vert             ; Target vertical scan line position
                                        public  Max_Horiz               ; Maximum horizontal pixel position
                                        public  Max_Vert                ; Maximum vertical scan line position
                                        public  Horiz_Rate              ; Horizontal Pixels/update
                                        public  Vert_Rate               ; Vertical Scan lines/update
```

```
Microsoft (R) Macro Assembler  Version 4.00              9/9/86 12:00:00

Smooth_Scroll   Sample Smooth Scrolling Routine          Page     1-2

                                        public  Delay                   ; Updates/Vertical frame
                                        public  Count                   ; Current delay count value
                                        public  Pan_Table               ; Correct panning table address
                                        public  Start_Address           ; Start address save area
                                        public  Preset_Pan              ; Preset row scan/Pixel pan save area
                                        public  Scroll_Flag             ; Scroll flag
                                        public  Vertical_End            ; Vertical end register
                                        extrn   Logical_Window:near     ; Logical windowing routine
                                        extrn   Present:near            ; Check enhanced presence routine
                                        extrn   Active:near             ; Check enhanced active routine
                                        extrn   Actual:near             ; Get actual mode value routine
                                        extrn   Input:near              ; Input register routine
                                        extrn   Output:near             ; Output register routine
                                        extrn   Out_Pair:near           ; Output register pair routine
                                        extrn   Wait_Vert:near          ; Wait for vertical retrace routine
                                        extrn   Wait_Active:near        ; Wait for active display routine
                                ;
                                ;  LOCAL Equates
                                ;
 = 0040                         BIOS_SEG        equ     40h             ; ROM BIOS data area segment
 = 0010                         VIDEO           equ     10h             ; BIOS video interrupt number
 = 000F                         GET_MODE        equ     0Fh             ; Get current video mode function code
 = 007F                         MODE_MASK       equ     7Fh             ; Current video mode mask value
 = 0007                         MONO_MODE       equ     07h             ; Monochrome video mode
 = 0006                         STAT_PORT       equ     06h             ; Offset to input status port one
 = 0008                         PRESET_ROW      equ     08h             ; CRT controller preset row scan reg.
```

SMOOTH SCROLLING AND PANNING

```
Microsoft (R) Macro Assembler  Version 4.00              9/9/86 12:00:00

Smooth_Scroll   Sample Smooth Scrolling Routine          Page     1-3


 = 000C                         START_HIGH      equu     0Ch            ; CRT controller start address MSW reg.
 = 000D                         START_LOW       equ     0Dh            ; CRT controller start address LSW reg.
 = 03C0                         ATTR_INDEX      equ     3C0h           ; Attribute controller index port
 = 0013                         PIXEL_PAN       equ     13h            ; Attribute controller pixel pan reg.
 = 0020                         PALETTE         equ     20h            ; Attribute controller palette source
 = 00FF                         SCROLL_NEEDED   equ     0FFh           ; Scroll needed flag value
 = 0001                         SCROLLING       equ     01h            ; Scrolling flag value
 = 0063                         v_base          equ     word ptr 063h  ; CRT controller base address offset
                                ;
                                ;  Define the code segment
                                ;
 0000                           code    segment byte public 'CODE'     ; Standard code segment
                                        assume  cs:code, ds:nothing
```

```
Microsoft (R) Macro Assembler  Version 4.00             9/9/86 12:00:00

Smooth_Scroll   Sample Smooth Scrolling Routine          Page     1-4


                                       page
                                ;**************************************************************************
                                ;
                                ;       Scroll_Setup()
                                ;
                                ;       Registers on Entry:
                                ;
                                ;               AH    - Number of screens horizontally
                                ;               AL    - Number of screens vertically
                                ;               CX    - Logical line width
                                ;
                                ;       Registers on Exit:
                                ;
                                ;               None
                                ;               CY    - Set if not on an enhanced graphics adapter
                                ;
                                ;**************************************************************************
0000                            Scroll_Setup    Proc    Near            ; Smooth scroll setup procedure
0000  50                                push    ax                      ; Save the AX register value
0001  53                                push    bx                      ; Save the BX register value
0002  51                                push    cx                      ; Save the CX register value
0003  52                                push    dx                      ; Save the DX register value
0004  2E: 89 0E 0254 R                  mov     cs:[Logical_Width],cx   ; Save the logical line width
0009  8B C8                             mov     cx,ax                   ; Save the logical screen size
000B  E8 0000 E                         call    Present                 ; Check for enhanced adapter present
000E  72 1E                             jc      Return_Setup            ; Jump if NO enhanced adapter present
```

```
Microsoft (R) Macro Assembler  Version 4.00                 9/9/86 12:00:00

Smooth_Scroll   Sample Smooth Scrolling Routine             Page     1-5


 0010  E8 0000 E                     call   Active                 ; Check for enhanced adapter active
 0013  72 19                         jc     Return_Setup           ; Jump if enhanced adapter NOT active
 0015  B4 0F                         mov    ah,GET_MODE            ; Get the BIOS get video mode function
 0017  CD 10                         int    VIDEO                  ; Get the current video mode value
 0019  24 7F                         and    al,MODE_MASK           ; Mask off all but the video mode
 001B  8D 1E 0352 R                  lea    bx,cs:[Dot_8]          ; Get offset of the 8 dot panning table
 001F  3C 07                         cmp    al,MONO_MODE           ; Check for monochrome mode
 0021  75 04                         jne    Save_Pan               ; Jump if NOT in monochrome mode
 0023  8D 1E 0349 R                  lea    bx,cs:[Dot_9]          ; Get offset of the 9 dot panning table
 0027                        Save_Pan:
 0027  2E: 89 1E 0272 R              mov    cs:[Pan_Table],bx      ; Save correct panning table address
 002C  EB 05                         jmp    SHORT Do_Setup         ; Go do the scroll setup
 002E                        Return_Setup:
 002E  5A                            pop    dx                     ; Restore the DX register value
 002F  59                            pop    cx                     ; Restore the CX register value
 0030  5B                            pop    bx                     ; Restore the BX register value
 0031  58                            pop    ax                     ; Restore the AX register value
 0032  C3                            ret                           ; Return to the caller
 0033                        Do_Setup:
 0033  E8 0000 E                     call   actual                 ; Call routine to get actual mode
 0036  98                            cbw                           ; Convert video mode to full word
 0037  8B D8                         mov    bx,ax                  ; Move the video mode value to BX
 0039  2E: 8A 87 027A R              mov    al,cs:[Horiz_Table+bx] ; Get the horizontal step size
 003E  2E: A3 0256 R                 mov    cs:[Unit_Horiz],ax     ; Save the pixels/horizontal unit
 0042  2E: 8A 87 0291 R              mov    al,cs:[Horiz_Step+bx]  ; Get the horizontal step size
 0047  2E: A3 025A R                 mov    cs:[Horiz_Size],ax     ; Save the pixels/horizontal step
```

```
Microsoft (R) Macro Assembler  Version 4.00              9/9/86 12:00:00

Smooth_Scroll   Sample Smooth Scrolling Routine          Page     1-6


 004B  2E: 8A 87 02A8 R            mov    al,cs:[Vert_Table+bx]   ; Get the vertical step size
 0050  2E: A3 0258 R               mov    cs:[Unit_Vert],ax       ; Save the scan lines/vertical unit
 0054  2E: 8A 87 02BF R            mov    al,cs:[Vert_Step+bx]    ; Get the vertical step size
 0059  2E: A3 025C R               mov    cs:[Vert_Size],ax       ; Save the scan lines/vertical step
 005D  2E: 8A 87 0332 R            mov    al,cs:[End_Table+bx]    ; Get the vertical retrace end register
 0062  2E: A2 0279 R               mov    cs:[Vertical_End],al    ; Save the vertical end register value
 0066  D1 E3                       shl    bx,1                    ; Convert to word table index
 0068  8A C5                       mov    al,ch                   ; Get number of screens horizontally
 006A  98                          cbw                            ; Convert screens to full word
 006B  48                          dec    ax                      ; Decrement for calculation
 006C  2E: F7 A7 02D6 R            mul    cs:[Pixel_Table+bx]     ; Calculate maximum horizontal value
 0071  2E: A3 0266 R               mov    cs:[Max_Horiz],ax       ; Save the maximum horizontal value
 0075  8A C1                       mov    al,cl                   ; Get number of screens vertically
 0077  98                          cbw                            ; Convert screens to full word
 0078  48                          dec    ax                      ; Decrement for calculation
 0079  2E: F7 A7 0304 R            mul    cs:[Scan_Table+bx]      ; Calculate maximum vertical value
 007E  2E: A3 0268 R               mov    cs:[Max_Vert],ax        ; Save the maximum vertical value
 0082  33 C0                       xor    ax,ax                   ; Setup to zero current/target location
 0084  2E: A3 025E R               mov    cs:[Current_Horiz],ax   ; Zero the current horizontal position
 0088  2E: A3 0260 R               mov    cs:[Current_Vert],ax    ; Zero the current vertical position
 008C  2E: A3 0262 R               mov    cs:[Target_Horiz],ax    ; Zero the target horizontal position
 0090  2E: A3 0264 R               mov    cs:[Target_Vert],ax     ; Zero the target vertical position
 0094  2E: A3 0274 R               mov    cs:[Start_Address],ax   ; Zero the current start address
 0098  2E: A3 0276 R               mov    cs:[Preset_Pan],ax      ; Zero preset row scan/pixel pan value
 009C  2E: A2 0278 R               mov    cs:[Scroll_Flag],al     ; Zero the scroll flag
 00A0  F8                          clc                            ; Clear carry indicating no error
```

```
Microsoft (R) Macro Assembler  Version 4.00                9/9/86 12:00:00

Smooth_Scroll   Sample Smooth Scrolling Routine            Page     1-7


 00A1  EB 8B                          jmp      Return_Setup            ; Go return to the caller
                              Scroll_Setup     Endp                    ; End of smooth scroll setup procedure
```

```
Microsoft (R) Macro Assembler  Version 4.00                 9/9/86 12:00:00

Smooth_Scroll   Sample Smooth Scrolling Routine             Page     1-8


                                        page
                                ;******************************************************************************
                                ;
                                ;       Smooth_Scroll(Horiz_Units, Vert_Units, Horiz_Rate, Vert_Rate, Delay)
                                ;
                                ;       Registers on Entry:
                                ;
                                ;               AH    - Number of horizontal units(Chars/Pixels) [-128 to +127]
                                ;               AL    - Number of vertical units(Rows/Scan lines)[-128 to +127]
                                ;               BH    - Horizontal rate[Zero=Current]           (Pixels/Update)
                                ;               BL    - Vertical rate  [Zero=Current]       (Scan lines/Update)
                                ;               CX    - Delay value    [Zero=Current]  (Vertical Frames/Update)
                                ;
                                ;       Registers on Exit:
                                ;
                                ;               None
                                ;
                                ;               CY    - Set if scroll outside logical window
                                ;
                                ;******************************************************************************
00A3                            Smooth_Scroll   Proc    Near            ; Smooth scrolling procedure
00A3  50                                push    ax                      ; Save the AX register value
00A4  53                                push    bx                      ; Save the BX register value
00A5  51                                push    cx                      ; Save the CX register value
00A6  52                                push    dx                      ; Save the DX register value
00A7  56                                push    si                      ; Save the SI register value
```

```
Microsoft (R) Macro Assembler  Version 4.00             9/9/86 12:00:00

Smooth_Scroll   Sample Smooth Scrolling Routine          Page     1-9


 00A8  57                        push    di                  ; Save the DI register value
 00A9  0B C9                     or      cx,cx               ; Check for delay given (Non-zero)
 00AB  74 05                     jz      Calculate           ; Jump if delay not given (Use current)
 00AD  2E: 89 0E 026E R          mov     cs:[Delay],cx       ; Save the delay value that was given
 00B2                    Calculate:
 00B2  8B C8                     mov     cx,ax               ; Move the movement values into CX
 00B4  8A C5                     mov     al,ch               ; Get the horizontal movement value
 00B6  98                        cbw                         ; Convert the byte to a full word
 00B7  2E: F7 2E 0256 R          imul    cs:[Unit_Horiz]     ; Multiply by pixels/horizontal unit
 00BC  2E: 03 06 0262 R          add     ax,cs:[Target_Horiz] ; Add in the current horizontal target
 00C1  2E: 3B 06 0266 R          cmp     ax,cs:[Max_Horiz]   ; Check for scroll out of range
 00C6  77 30                     ja      Scroll_Error        ; Jump if the scroll is out of range
 00C8  8B F0                     mov     si,ax               ; Save the new target value in SI
 00CA  8A C1                     mov     al,cl               ; Get the vertical movement value
 00CC  98                        cbw                         ; Convert the byte to a full word
 00CD  2E: F7 2E 0258 R          imul    cs:[Unit_Vert]      ; Multiply by scan lines/vertical unit
 00D2  2E: 03 06 0264 R          add     ax,cs:[Target_Vert] ; Add in the current vertical target
 00D7  2E: 3B 06 0268 R          cmp     ax,cs:[Max_Vert]    ; Check for scroll out of range
 00DC  77 1A                     ja      Scroll_Error        ; Jump if the scroll is out of range
 00DE  8B F8                     mov     di,ax               ; Save the new target value in DI
 00E0  32 E4                     xor     ah,ah               ; Zero the upper half of AX register
 00E2  0A FF                     or      bh,bh               ; Check for no given value (Zero)
 00E4  74 06                     jz      Check_Vert          ; Jump if no given value (Use current)
 00E6  8A C7                     mov     al,bh               ; Get the given horizontal rate (Word)
 00E8  2E: A3 026A R             mov     cs:[Horiz_Rate],ax  ; Save the given horizontal rate
 00EC                    Check_Vert:
```

```
Microsoft (R) Macro Assembler  Version 4.00               9/9/86 12:00:00

Smooth_Scroll   Sample Smooth Scrolling Routine           Page     1-10


 00EC  0A DB                     or     bl,bl                  ; Check for no given value (Zero)
 00EE  74 0C                     jz     Scroll_Ready           ; Jump if no given value (Use current)
 00F0  8A C3                     mov    al,bl                  ; Get the given vertical rate (Word)
 00F2  2E: A3 026C R             mov    cs:[Vert_Rate],ax      ; Save the given vertical rate
 00F6  EB 04                     jmp    SHORT Scroll_Ready     ; Ready to go do the scroll
 00F8                    Scroll_Error:
 00F8  F9                        stc                           ; Scroll outside limits, set carry flag
 00F9  EB 28 90                  jmp    Scroll_Exit            ; Go return to caller with error
 00FC                    Scroll_Ready:
 00FC  FA                        cli                           ;;; Disable interrupts (Lockout)
 00FD  2E: 89 36 0262 R          mov    cs:[Target_Horiz],si   ;;; Update the horizontal target value
 0102  2E: 89 3E 0264 R          mov    cs:[Target_Vert],di    ;;; Update the vertical target value
 0107  2E: A0 0278 R             mov    al,cs:[Scroll_Flag]    ;;; Get the scroll flag value
 010B  0A C0                     or     al,al                  ;;; Check for already scrolling
 010D  75 12                     jnz    All_Set                ;;; Jump if already scrolling
 010F  2E: C6 06 0278 R FF       mov    Byte Ptr cs:[Scroll_Flag],SCROLL_NEEDED
 0115  2E: A1 026E R             mov    ax,cs:[Delay]          ;;; Get the desired delay value
 0119  2E: A3 0270 R             mov    cs:[Count],ax          ;;; Update the current count value
 011D  FB                        sti                           ;;; Enable interrupts (Unlock)
 011E  E8 012A R                 call   Do_Scroll              ; Call routine to do the scroll
 0121                    All_Set:
 0121  FB                        sti                           ;;; Enable interrupts (Unlock)
 0122  F8                        clc                           ; Clear carry indicating no error
 0123                    Scroll_Exit:
 0123  5F                        pop    di                     ; Restore the DI register value
 0124  5E                        pop    si                     ; Restore the SI register value
```

```
Microsoft (R) Macro Assembler  Version 4.00              9/9/86 12:00:00

Smooth_Scroll   Sample Smooth Scrolling Routine          Page     1-11


 0125  5A                           pop     dx                      ; Restore the DX register value
 0126  59                           pop     cx                      ; Restore the CX register value
 0127  5B                           pop     bx                      ; Restore the BX register value
 0128  58                           pop     ax                      ; Restore the AX register value
 0129  C3                           ret                             ; Return to the caller
                            Smooth_Scroll   Endp                    ; End of smooth scrolling procedure
```

```
Microsoft (R) Macro Assembler  Version 4.00             9/9/86 12:00:00

Smooth_Scroll   Sample Smooth Scrolling Routine          Page     1-12


                                    page
                                ;*******************************************************************************
                                ;
                                ;       Do_Scroll()
                                ;
                                ;       Registers on Entry:
                                ;
                                ;               None
                                ;
                                ;               The following variables MUST be set up on entry
                                ;
                                ;                       Target_Horiz  - Target horizontal pixel
                                ;                       Target_Vert   - Target vertical scan line
                                ;                       Scroll_Flag   - Scroll flag  (Scrolling or Needed)
                                ;                       Horiz_Rate    - Pixels/Update
                                ;                       Horiz_Size    - Pixels/Horizontal step  (Physical)
                                ;                       Vert_Rate     - Scan lines/Update
                                ;                       Vert_Size     - Scan lines/Vertical step (Physical)
                                ;                       Delay         - Vertical Frames/Update
                                ;
                                ;       Registers on Exit:
                                ;
                                ;               None
                                ;
                                ;*******************************************************************************
 012A                           Do_Scroll       Proc    Near            ; Do the scroll procedure
```

```
Microsoft (R) Macro Assembler  Version 4.00                 9/9/86 12:00:00

Smooth_Scroll   Sample Smooth Scrolling Routine             Page     1-13


 012A  50                          push    ax                  ; Save the AX register value
 012B  53                          push    bx                  ; Save the BX register value
 012C  51                          push    cx                  ; Save the CX register value
 012D  52                          push    dx                  ; Save the DX register value
 012E  1E                          push    ds                  ; Save the DS register value
 012F  B8 0040                     mov     ax,BIOS_SEG         ; Get the BIOS data area segment
 0132  8E D8                       mov     ds,ax               ; Setup to access the BIOS data area
 0134                         Scroll_Loop:
 0134  2E: A0 0278 R               mov     al,cs:[Scroll_Flag] ; Get the current scroll flag
 0138  0A C0                       or      al,al               ; Check for more scrolling needed
 013A  74 1A                       jz      Scroll_Done         ; Jump if no more scrolling needed
 013C  FE C0                       inc     al                  ; Check for scrolling just started
 013E  74 60                       jz      Setup_Next          ; Jump if scrolling just started
 0140  8B 16 0063                  mov     dx,ds:[v_base]      ; Get the CRT base port address
 0144  83 C2 06                    add     dx,STAT_PORT        ; Increment to the status port address
 0147                         Wait_Loop:
 0147  E8 0000 E                   call    Wait_Active         ; Wait for an active display time
 014A  2E: FF 0E 0270 R            dec     Word Ptr cs:[Count] ; Decrement the current delay count
 014F  74 0B                       jz      Do_Update           ; Jump if time to do an update
 0151  E8 0000 E                   call    Wait_vert           ; Wait for end of this vertical frame
 0154  EB F1                       jmp     Wait_Loop           ; Loop till it is time to update
 0156                         Scroll_Done:
 0156  1F                          pop     ds                  ; Restore the DS register value
 0157  5A                          pop     dx                  ; Restore the DX register value
 0158  59                          pop     cx                  ; Restore the CX register value
 0159  5B                          pop     bx                  ; Restore the BX register value
```

```
Microsoft (R) Macro Assembler  Version 4.00             9/9/86 12:00:00

Smooth_Scroll   Sample Smooth Scrolling Routine         Page     1-14


 015A  58                        pop    ax                    ; Restore the AX register value
 015B  C3                        ret                          ; Return to the caller
 015C                    Do_Update:
 015C  2E: A1 026E R             mov    ax,cs:[Delay]         ; Get the desired delay count
 0160  2E: A3 0270 R             mov    cs:[Count],ax         ; Reset the current count value
 0164  2E: 8B 1E 0274 R          mov    bx,cs:[Start_Address] ; Get the new start address
 0169  2E: 8B 0E 0276 R          mov    cx,cs:[Preset_Pan]    ; Get the new preset and panning values
 016E  83 EA 06                  sub    dx,STAT_PORT          ; Decrement back to CRT base port
 0171  B4 0C                     mov    ah,START_HIGH         ; Get the start address high register
 0173  8A C7                     mov    al,bh                 ; Get the MSB of the new start address
 0175  E8 0000 E                 call   Out_Pair              ; Output new start address high value
 0178  B4 0D                     mov    ah,START_LOW          ; Get the start address low register
 017A  8A C3                     mov    al,bl                 ; Get the LSB of the new start address
 017C  E8 0000 E                 call   Out_Pair              ; Output new start address low value
 017F  B4 08                     mov    ah,PRESET_ROW         ; Get the preset row scan register
 0181  8A C5                     mov    al,ch                 ; Get the new preset row scan value
 0183  E8 0000 E                 call   Out_Pair              ; Output the new preset row scan value
 0186  83 C2 06                  add    dx,STAT_PORT          ; Increment back to the status port
 0189  E8 0000 E                 call   Wait_Vert             ;;; Wait for vertical retrace period
 018C  E8 0000 E                 call   Input                 ;;; Reset the attribute flip-flop
 018F  BA 03C0                   mov    dx,ATTR_INDEX         ;;; Get the attribute controller port
 0192  B4 33                     mov    ah,PIXEL_PAN+PALETTE  ;;; Get the pixel pan register index
 0194  8A C1                     mov    al,cl                 ;;; Get the new pixel panning value
 0196  E8 0000 E                 call   Out_Pair              ;;; Output the new pixel panning value
 0199  FB                        sti                          ;;; Enable interrupts again (Unlock)
 019A  32 C0                     xor    al,al                 ; Setup to reset the scroll flag
```

```
Microsoft (R) Macro Assembler  Version 4.00              9/9/86 12:00:00

Smooth_Scroll   Sample Smooth Scrolling Routine          Page     1-15


 019C  2E: A2 0278 R                  mov    cs:[Scroll_Flag],al      ; Reset the scroll flag value
 01A0                            Setup_Next:
 01A0  2E: A1 0260 R                  mov    ax,cs:[Current_Vert]     ; Get the current vertical scan line
 01A4  2E: 3B 06 0264 R               cmp    ax,cs:[Target_Vert]      ; Compare against the target scan line
 01A9  74 34                          je     Do_Horiz                 ; Jump if at the target value
 01AB  72 18                          jb     Scroll_Down              ; Jump if below target value
 01AD                            Scroll_Up:
 01AD  2E: C6 06 0278 R 01            mov    Byte Ptr cs:[Scroll_Flag],SCROLLING
 01B3  2E: 2B 06 026C R               sub    ax,cs:[Vert_Rate]        ; Subtract the vertical rate value
 01B8  2E: 3B 06 0264 R               cmp    ax,cs:[Target_Vert]      ; Compare against target value again
 01BD  7D 1C                          jge    Set_Vert                 ; Jump if not past the target value
 01BF  2E: A1 0264 R                  mov    ax,cs:[Target_Vert]      ; Past target, use target value instead
 01C3  EB 16                          jmp    SHORT Set_Vert           ; Go save the new vertical scan line
 01C5                            Scroll_Down:
 01C5  2E: C6 06 0278 R 01            mov    Byte Ptr cs:[Scroll_Flag],SCROLLING
 01CB  2E: 03 06 026C R               add    ax,cs:[Vert_Rate]        ; Add the vertical rate value
 01D0  2E: 3B 06 0264 R               cmp    ax,cs:[Target_Vert]      ; Compare against target value again
 01D5  7E 04                          jle    Set_Vert                 ; Jump if not past the target value
 01D7  2E: A1 0264 R                  mov    ax,cs:[Target_Vert]      ; Past target, use target value instead
 01DB                            Set_Vert:
 01DB  2E: A3 0260 R                  mov    cs:[Current_Vert],ax     ; Update the current vertical scan line
 01DF                            Do_Horiz:
 01DF  2E: A1 025E R                  mov    ax,cs:[Current_Horiz]    ; Get the current horizontal pixel
 01E3  2E: 3B 06 0262 R               cmp    ax,cs:[Target_Horiz]     ; Compare against the target pixel
 01E8  74 34                          je     Do_Calculate             ; Jump if at the target value
 01EA  72 18                          jb     Scroll_Right             ; Jump if below the target value
```

```
Microsoft (R) Macro Assembler  Version 4.00            9/9/86 12:00:00

Smooth_Scroll   Sample Smooth Scrolling Routine         Page     1-16


 01EC                         Scroll_Left:
 01EC  2E: C6 06 0278 R 01            mov     Byte Ptr cs:[Scroll_Flag],SCROLLING
 01F2  2E: 2B 06 026A R               sub     ax,cs:[Horiz_Rate]      ; Subtract the horizontal rate value
 01F7  2E: 3B 06 0262 R               cmp     ax,cs:[Target_Horiz]    ; Compare against target value again
 01FC  7D 1C                          jge     Set_Horiz               ; Jump if not past the target value
 01FE  2E: A1 0262 R                  mov     ax,cs:[Target_Horiz]    ; Past target, use target value instead
 0202  EB 16                          jmp     SHORT Set_Horiz         ; Go save the new horizontal pixel
 0204                         Scroll_Right:
 0204  2E: C6 06 0278 R 01            mov     Byte Ptr cs:[Scroll_Flag],SCROLLING
 020A  2E: 03 06 026A R               add     ax,cs:[Horiz_Rate]      ; Add the horizontal rate value
 020F  2E: 3B 06 0262 R               cmp     ax,cs:[Target_Horiz]    ; Compare against target value again
 0214  7E 04                          jle     Set_Horiz               ; Jump if not past the target value
 0216  2E: A1 0262 R                  mov     ax,cs:[Target_Horiz]    ; Past target, use target value instead
 021A                         Set_Horiz:
 021A  2E: A3 025E R                  mov     cs:[Current_Horiz],ax   ; Update the current horizontal pixel
 021E                         Do_Calculate:
 021E  2E: A1 0260 R                  mov     ax,cs:[Current_Vert]    ; Get the current vertical scan line
 0222  99                             cwd                             ; Convert value to a double word
 0223  2E: F7 36 025C R               div     cs:[Vert_Size]          ; Divide by the vertical step size
 0228  8A EA                          mov     ch,dl                   ; Save preset row scan value (Remainder)
 022A  2E: F7 26 0254 R               mul     cs:[Logical_Width]      ; Multiply by logical line length
 022F  8B D8                          mov     bx,ax                   ; Save result as new offset value
 0231  2E: A1 025E R                  mov     ax,cs:[Current_Horiz]   ; Get the current horizontal pixel
 0235  99                             cwd                             ; Convert value to a double word
 0236  2E: F7 36 025A R               div     cs:[Horiz_Size]         ; Divide by the horizontal step size
 023B  8B F2                          mov     si,dx                   ; Save the pixel shift in SI (Remainder)
```

```
Microsoft (R) Macro Assembler  Version 4.00              9/9/86 12:00:00

Smooth_Scroll   Sample Smooth Scrolling Routine          Page     1-17


 023D  2E: 8B 2E 0272 R                mov     bp,cs:[Pan_Table]        ; Get the correct panning table address
 0242  2E: 8A 0A                       mov     cl,cs:[bp+si]            ; Save the pixel panning value (Lookup)
 0245  03 D8                           add     bx,ax                    ; Add result to get new offset value
 0247  2E: 89 1E 0274 R                mov     cs:[Start_Address],bx    ; Save the new start address value
 024C  2E: 89 0E 0276 R                mov     cs:[Preset_Pan],cx       ; Save the new preset/pan values
 0251  E9 0134 R                       jmp     Scroll_Loop              ; Go do more scrolling if needed
                               Do_Scroll       Endp                     ; End of the do scroll procedure
                               ;
                               ;  Define the local data values
                               ;
 0254  0000                    Logical_Width   dw      0                ; Logical line width
 0256  0000                    Unit_Horiz      dw      0                ; Pixels/horizontal unit
 0258  0000                    Unit_Vert       dw      0                ; Scan lines/vertical unit
 025A  0000                    Horiz_Size      dw      0                ; Pixels/horizontal step  (Physical)
 025C  0000                    Vert_Size       dw      0                ; Scan lines/vertical step (Physical)
 025E  0000                    Current_Horiz   dw      0                ; Current horizontal pixel position
 0260  0000                    Current_Vert    dw      0                ; Current vertical scan line position
 0262  0000                    Target_Horiz    dw      0                ; Target horizontal pixel position
 0264  0000                    Target_Vert     dw      0                ; Target vertical scan line position
 0266  0000                    Max_Horiz       dw      0                ; Maximum horizontal pixel position
 0268  0000                    Max_Vert        dw      0                ; Maximum vertical scan line position
 026A  0001                    Horiz_Rate      dw      1                ; Horizontal Pixels/update
 026C  0001                    Vert_Rate       dw      1                ; Vertical Scan lines/update
 026E  0001                    Delay           dw      1                ; Updates/Vertical frame
 0270  0000                    Count           dw      0                ; Current delay count value
 0272  0000                    Pan_Table       dw      0                ; Correct panning table address
```

```
Microsoft (R) Macro Assembler  Version 4.00              9/9/86 12:00:00

Smooth_Scroll   Sample Smooth Scrolling Routine          Page     1-18


 0274  0000                 Start_Address   dw      0               ; Start address save area
 0276  0000                 Preset_Pan      dw      0               ; Preset row scan/Pixel pan save area
 0278  00                   Scroll_Flag     db      0               ; Scroll flag
 0279  00                   Vertical_End    db      0               ; Vertical end register
                            ;
                            ;  Define the pixels/horizontal unit table
                            ;
 = 027A                     Horiz_Table     equ     This Byte       ; Pixels/Horizontal unit table
 027A  08                                   db      08h             ; Mode 0   Pixels/Horizontal Unit
 027B  08                                   db      08h             ; Mode 1   Pixels/Horizontal Unit
 027C  08                                   db      08h             ; Mode 2   Pixels/Horizontal Unit
 027D  08                                   db      08h             ; Mode 3   Pixels/Horizontal Unit
 027E  01                                   db      01h             ; Mode 4   Pixels/Horizontal Unit
 027F  01                                   db      01h             ; Mode 5   Pixels/Horizontal Unit
 0280  01                                   db      01h             ; Mode 6   Pixels/Horizontal Unit
 0281  09                                   db      09h             ; Mode 7   Pixels/Horizontal Unit
 0282  08                                   db      08h             ; Mode 8   Pixels/Horizontal Unit
 0283  08                                   db      08h             ; Mode 9   Pixels/Horizontal Unit
 0284  08                                   db      08h             ; Mode A   Pixels/Horizontal Unit
 0285  08                                   db      08h             ; Mode B   Pixels/Horizontal Unit
 0286  09                                   db      09h             ; Mode C   Pixels/Horizontal Unit
 0287  01                                   db      01h             ; Mode D   Pixels/Horizontal Unit
 0288  01                                   db      01h             ; Mode E   Pixels/Horizontal Unit
 0289  01                                   db      01h             ; Mode F   Pixels/Horizontal Unit
 028A  01                                   db      01h             ; Mode 10  Pixels/Horizontal Unit
 028B  01                                   db      01h             ; Mode F+  Pixels/Horizontal Unit
```

```
Microsoft (R) Macro Assembler  Version 4.00              9/9/86 12:00:00

Smooth_Scroll   Sample Smooth Scrolling Routine           Page     1-19


 028C  01                               db      01h            ; Mode 10+ Pixels/Horizontal Unit
 028D  08                               db      08h            ; Mode 0*  Pixels/Horizontal Unit
 028E  08                               db      08h            ; Mode 1*  Pixels/Horizontal Unit
 028F  08                               db      08h            ; Mode 2*  Pixels/Horizontal Unit
 0290  08                               db      08h            ; Mode 3*  Pixels/Horizontal Unit
                           ;
                           ;  Define the horizontal step size table
                           ;
 = 0291                    Horiz_Step   equ     This Byte      ; Horizontal step size table
 0291  08                               db      08h            ; Mode 0   Pixels/Horizontal Step
 0292  08                               db      08h            ; Mode 1   Pixels/Horizontal Step
 0293  08                               db      08h            ; Mode 2   Pixels/Horizontal Step
 0294  08                               db      08h            ; Mode 3   Pixels/Horizontal Step
 0295  04                               db      04h            ; Mode 4   Pixels/Horizontal Step
 0296  04                               db      04h            ; Mode 5   Pixels/Horizontal Step
 0297  08                               db      08h            ; Mode 6   Pixels/Horizontal Step
 0298  09                               db      09h            ; Mode 7   Pixels/Horizontal Step
 0299  08                               db      08h            ; Mode 8   Pixels/Horizontal Step
 029A  08                               db      08h            ; Mode 9   Pixels/Horizontal Step
 029B  08                               db      08h            ; Mode A   Pixels/Horizontal Step
 029C  08                               db      08h            ; Mode B   Pixels/Horizontal Step
 029D  09                               db      09h            ; Mode C   Pixels/Horizontal Step
 029E  08                               db      08h            ; Mode D   Pixels/Horizontal Step
 029F  08                               db      08h            ; Mode E   Pixels/Horizontal Step
 02A0  08                               db      08h            ; Mode F   Pixels/Horizontal Step
 02A1  08                               db      08h            ; Mode 10  Pixels/Horizontal Step
```

```
Microsoft (R) Macro Assembler  Version 4.00           9/9/86 12:00:00

Smooth_Scroll   Sample Smooth Scrolling Routine        Page     1-20

 02A2  08                           db     08h          ; Mode F+  Pixels/Horizontal Step
 02A3  08                           db     08h          ; Mode 10+ Pixels/Horizontal Step
 02A4  08                           db     08h          ; Mode 0*  Pixels/Horizontal Step
 02A5  08                           db     08h          ; Mode 1*  Pixels/Horizontal Step
 02A6  08                           db     08h          ; Mode 2*  Pixels/Horizontal Step
 02A7  08                           db     08h          ; Mode 3*  Pixels/Horizontal Step
                        ;
                        ;  Define the scan lines/vertical unit table
                        ;
 = 02A8                 Vert_Table  equ    This Byte    ; Scan lines/Vertical unit table
 02A8  08                           db     08h          ; Mode 0   Scan lines/Vertical Unit
 02A9  08                           db     08h          ; Mode 1   Scan lines/Vertical Unit
 02AA  08                           db     08h          ; Mode 2   Scan lines/Vertical Unit
 02AB  08                           db     08h          ; Mode 3   Scan lines/Vertical Unit
 02AC  01                           db     01h          ; Mode 4   Scan lines/Vertical Unit
 02AD  01                           db     01h          ; Mode 5   Scan lines/Vertical Unit
 02AE  01                           db     01h          ; Mode 6   Scan lines/Vertical Unit
 02AF  0E                           db     0Eh          ; Mode 7   Scan lines/Vertical Unit
 02B0  08                           db     08h          ; Mode 8   Scan lines/Vertical Unit
 02B1  08                           db     08h          ; Mode 9   Scan lines/Vertical Unit
 02B2  08                           db     08h          ; Mode A   Scan lines/Vertical Unit
 02B3  08                           db     08h          ; Mode B   Scan lines/Vertical Unit
 02B4  0E                           db     0Eh          ; Mode C   Scan lines/Vertical Unit
 02B5  01                           db     01h          ; Mode D   Scan lines/Vertical Unit
 02B6  01                           db     01h          ; Mode E   Scan lines/Vertical Unit
 02B7  01                           db     01h          ; Mode F   Scan lines/Vertical Unit
```

```
Microsoft (R) Macro Assembler  Version 4.00               9/9/86 12:00:00

Smooth_Scroll   Sample Smooth Scrolling Routine           Page     1-21


 02B8  01                                 db      01h          ; Mode 10  Scan lines/Vertical Unit
 02B9  01                                 db      01h          ; Mode F+  Scan lines/Vertical Unit
 02BA  01                                 db      01h          ; Mode 10+ Scan lines/Vertical Unit
 02BB  0E                                 db      0Eh          ; Mode 0*  Scan lines/Vertical Unit
 02BC  0E                                 db      0Eh          ; Mode 1*  Scan lines/Vertical Unit
 02BD  0E                                 db      0Eh          ; Mode 2*  Scan lines/Vertical Unit
 02BE  0E                                 db      0Eh          ; Mode 3*  Scan lines/Vertical Unit
                           ;
                           ;  Define the vertical step size table
                           ;
 = 02BF                    Vert_Step      equ     This Byte    ; Vertical step size table
 02BF  08                                 db      08h          ; Mode 0   Scan lines/Vertical Step
 02C0  08                                 db      08h          ; Mode 1   Scan lines/Vertical Step
 02C1  08                                 db      08h          ; Mode 2   Scan lines/Vertical Step
 02C2  08                                 db      08h          ; Mode 3   Scan lines/Vertical Step
 02C3  01                                 db      01h          ; Mode 4   Scan lines/Vertical Step
 02C4  01                                 db      01h          ; Mode 5   Scan lines/Vertical Step
 02C5  01                                 db      01h          ; Mode 6   Scan lines/Vertical Step
 02C6  0E                                 db      0Eh          ; Mode 7   Scan lines/Vertical Step
 02C7  08                                 db      08h          ; Mode 8   Scan lines/Vertical Step
 02C8  08                                 db      08h          ; Mode 9   Scan lines/Vertical Step
 02C9  08                                 db      08h          ; Mode A   Scan lines/Vertical Step
 02CA  08                                 db      08h          ; Mode B   Scan lines/Vertical Step
 02CB  0E                                 db      0Eh          ; Mode C   Scan lines/Vertical Step
 02CC  01                                 db      01h          ; Mode D   Scan lines/Vertical Step
 02CD  01                                 db      01h          ; Mode E   Scan lines/Vertical Step
 02CE  01                                db      01h         ; Mode F   Scan lines/Vertical Step
 02CF  01                                db      01h         ; Mode 10  Scan lines/Vertical Step
 02D0  01                                db      01h         ; Mode F+  Scan lines/Vertical Step
 02D1  01                                db      01h         ; Mode 10+ Scan lines/Vertical Step
 02D2  0E                                db      0Eh         ; Mode 0*  Scan lines/Vertical Step
 02D3  0E                                db      0Eh         ; Mode 1*  Scan lines/Vertical Step
 02D4  0E                                db      0Eh         ; Mode 2*  Scan lines/Vertical Step
 02D5  0E                                db      0Eh         ; Mode 3*  Scan lines/Vertical Step
                             ;
                             ;  Define the pixels/frame table
                             ;
 = 02D6                      Pixel_Table equ     This Word   ; Pixels/Horizontal frame table
 02D6  0140                              dw      320         ; Mode 0   Pixels/Horizontal Frame
 02D8  0140                              dw      320         ; Mode 1   Pixels/Horizontal Frame
 02DA  0280                              dw      640         ; Mode 2   Pixels/Horizontal Frame
 02DC  0280                              dw      640         ; Mode 3   Pixels/Horizontal Frame
 02DE  0140                              dw      320         ; Mode 4   Pixels/Horizontal Frame
 02E0  0140                              dw      320         ; Mode 5   Pixels/Horizontal Frame
 02E2  0280                              dw      640         ; Mode 6   Pixels/Horizontal Frame
 02E4  02D0                              dw      720         ; Mode 7   Pixels/Horizontal Frame
 02E6  0140                              dw      320         ; Mode 8   Pixels/Horizontal Frame
 02E8  0140                              dw      320         ; Mode 9   Pixels/Horizontal Frame
 02EA  0140                              dw      320         ; Mode A   Pixels/Horizontal Frame
 02EC  0280                              dw      640         ; Mode B   Pixels/Horizontal Frame
 02EE  02D0                              dw      720         ; Mode C   Pixels/Horizontal Frame
 02F0  0140                              dw      320         ; Mode D   Pixels/Horizontal Frame
```

```
Microsoft (R) Macro Assembler  Version 4.00              9/9/86 12:00:00

Smooth_Scroll   Sample Smooth Scrolling Routine          Page     1-23


 02F2  0280                            dw      640          ; Mode E   Pixels/Horizontal Frame
 02F4  0280                            dw      640          ; Mode F   Pixels/Horizontal Frame
 02F6  0280                            dw      640          ; Mode 10  Pixels/Horizontal Frame
 02F8  0280                            dw      640          ; Mode F+  Pixels/Horizontal Frame
 02FA  0280                            dw      640          ; Mode 10+ Pixels/Horizontal Frame
 02FC  0140                            dw      320          ; Mode 0*  Pixels/Horizontal Frame
 02FE  0140                            dw      320          ; Mode 1*  Pixels/Horizontal Frame
 0300  0280                            dw      640          ; Mode 2*  Pixels/Horizontal Frame
 0302  0280                            dw      640          ; Mode 3*  Pixels/Horizontal Frame
                             ;
                             ;  Define the scan lines/frame table
                             ;
 = 0304                      Scan_Table equ     This Word    ; Scan lines/Vertical frame table
 0304  00C8                            dw      200          ; Mode 0   Scan lines/Vertical Frame
 0306  00C8                            dw      200          ; Mode 1   Scan lines/Vertical Frame
 0308  00C8                            dw      200          ; Mode 2   Scan lines/Vertical Frame
 030A  00C8                            dw      200          ; Mode 3   Scan lines/Vertical Frame
 030C  00C8                            dw      200          ; Mode 4   Scan lines/Vertical Frame
 030E  00C8                            dw      200          ; Mode 5   Scan lines/Vertical Frame
 0310  00C8                            dw      200          ; Mode 6   Scan lines/Vertical Frame
 0312  015E                            dw      350          ; Mode 7   Scan lines/Vertical Frame
 0314  00C8                            dw      200          ; Mode 8   Scan lines/Vertical Frame
 0316  00C8                            dw      200          ; Mode 9   Scan lines/Vertical Frame
 0318  00C8                            dw      200          ; Mode A   Scan lines/Vertical Frame
 031A  00C8                            dw      200          ; Mode B   Scan lines/Vertical Frame
 031C  015E                            dw      350          ; Mode C   Scan lines/Vertical Frame
```

```
Microsoft (R) Macro Assembler  Version 4.00              9/9/86 12:00:00

Smooth_Scroll   Sample Smooth Scrolling Routine          Page     1-24


 031E  00C8                          dw     200         ; Mode D   Scan lines/Vertical Frame
 0320  00C8                          dw     200         ; Mode E   Scan lines/Vertical Frame
 0322  015E                          dw     350         ; Mode F   Scan lines/Vertical Frame
 0324  015E                          dw     350         ; Mode 10  Scan lines/Vertical Frame
 0326  015E                          dw     350         ; Mode F+  Scan lines/Vertical Frame
 0328  015E                          dw     350         ; Mode 10+ Scan lines/Vertical Frame
 032A  015E                          dw     350         ; Mode 0*  Scan lines/Vertical Frame
 032C  015E                          dw     350         ; Mode 1*  Scan lines/Vertical Frame
 032E  015E                          dw     350         ; Mode 2*  Scan lines/Vertical Frame
 0330  015E                          dw     350         ; Mode 3*  Scan lines/Vertical Frame
                        ;
                        ;  Define the vertical end register table
                        ;
 = 0332                 End_Table    equu   This Byte   ; Vertical end register table
 0332  04                            db     04h         ; Mode 0   Vertical end register
 0333  04                            db     04h         ; Mode 1   Vertical end register
 0334  04                            db     04h         ; Mode 2   Vertical end register
 0335  04                            db     04h         ; Mode 3   Vertical end register
 0336  04                            db     04h         ; Mode 4   Vertical end register
 0337  04                            db     04h         ; Mode 5   Vertical end register
 0338  03                            db     03h         ; Mode 6   Vertical end register
 0339  0E                            db     0Eh         ; Mode 7   Vertical end register
 033A  04                            db     04h         ; Mode 8   Vertical end register
 033B  04                            db     04h         ; Mode 9   Vertical end register
 033C  04                            db     04h         ; Mode A   Vertical end register
 033D  04                            db     04h         ; Mode B   Vertical end register
```

```
Microsoft (R) Macro Assembler  Version 4.00               9/9/86 12:00:00

Smooth_Scroll   Sample Smooth Scrolling Routine           Page     1-25


 033E  0E                                   db      0Eh          ; Mode C   Vertical end register
 033F  04                                   db      04h          ; Mode D   Vertical end register
 0340  03                                   db      03h          ; Mode E   Vertical end register
 0341  0E                                   db      0Eh          ; Mode F   Vertical end register
 0342  0B                                   db      0Bh          ; Mode 10  Vertical end register
 0343  0E                                   db      0Eh          ; Mode F+  Vertical end register
 0344  0B                                   db      0Bh          ; Mode 10+ Vertical end register
 0345  0B                                   db      0Bh          ; Mode 0*  Vertical end register
 0346  0B                                   db      0Bh          ; Mode 1*  Vertical end register
 0347  0B                                   db      0Bh          ; Mode 2*  Vertical end register
 0348  0B                                   db      0Bh          ; Mode 3*  Vertical end register
                        ;
                        ;  Define the 9-Dot panning table
                        ;
 = 0349                 Dot_9               equ     This Byte    ; 9 Dot mode panning table
 0349  08                                   db      8            ; 0 Pixel shift value
 034A  00                                   db      0            ; 1 Pixel shift value
 034B  01                                   db      1            ; 2 Pixel shift value
 034C  02                                   db      2            ; 3 Pixel shift value
 034D  03                                   db      3            ; 4 Pixel shift value
 034E  04                                   db      4            ; 5 Pixel shift value
 034F  05                                   db      5            ; 6 Pixel shift value
 0350  06                                   db      6            ; 7 Pixel shift value
 0351  07                                   db      7            ; 8 Pixel shift value
                        ;
                        ;  Define the 8-Dot panning table
```

```
Microsoft (R) Macro Assembler  Version 4.00            9/9/86 12:00:00

Smooth_Scroll   Sample Smooth Scrolling Routine        Page     1-26


                               ;
 = 0352                        Dot_8        equ     This Byte       ; 8 Dot mode panning table
 0352  00                                   db      0               ; 0 Pixel shift value
 0353  01                                   db      1               ; 1 Pixel shift value
 0354  02                                   db      2               ; 2 Pixel shift value
 0355  03                                   db      3               ; 3 Pixel shift value
 0356  04                                   db      4               ; 4 Pixel shift value
 0357  05                                   db      5               ; 5 Pixel shift value
 0358  06                                   db      6               ; 6 Pixel shift value
 0359  07                                   db      7               ; 7 Pixel shift value
                               ;
                               ;  Define the end of the code segment
                               ;
 035A                          code    ends
                                       end                          ; End of the smooth scrolling module
```

B.6 USING THE VERTICAL BLANKING INTERRUPT

The vertical blanking interrupt is provided to allow software to alter a portion of the screen (such as a mouse cursor) and/or update display registers (such as those that control logical windowing, smooth scrolling, or panning). This interrupt, IRQ2, signals the onset of the vertical blanking interval, which is the longest contiguous time available for software to affect the screen without flicker. It also provides an event needed for synchronously advancing the registers used for smooth scrolling and panning, and for general animation timing.

Table B-1 indicates how much time is available in the vertical blanking interval, given the BIOS mode and type of monitor attached.

Table B-1. Vertical Blanking Interval

Mode	Monitor	Time
0..6, D..E	COMPAQ Color	3820 us
	COMPAQ Dual-Mode	3820 us
	RGBI Color	3820 us
0*..3*, 10+	COMPAQ Color	640 us
	COMPAQ Dual-Mode	970 us
7, F+	COMPAQ Dual-Mode	970 us

When the COMPAQ Enhanced Color Graphics Board is installed in a 80286- or 80386-based computer, IRQ9 is used instead of IRQ2. IRQ9 is directed to INT 71h; however, the system BIOS default handler re-routes control to INT 0Ah for compatibility with 8088- and 8086-based hardware. If the 80286 or 80386 is operated in the protected or virtual modes, INT 0Ah then becomes the INVALID TSS interrupt and operating system software can intercept vertical blanking interrupts directly via INT 71h.

```
Microsoft (R) Macro Assembler  Version 4.00            9/9/86 12:00:00

Vertical_INT    Sample Vertical Interrupt Routine        Page     1-1


                page    30,120
                title   Vertical_INT    Sample Vertical Interrupt Routine
;*******************************************************************************
;
;  Module Functional Description:
;
;               This module contains routines to perform smooth scrolling
;       on a logical window that is greater than the physical window (screen)
;       using the vertical retrace interrupt feature.
;
;*******************************************************************************
        public  Perform_Scroll          ; Smooth scrolling routine
        public  Vert_INT                ; Vertical interrupt routine
        extrn   Input:near              ; Input register routine
        extrn   Output:near             ; Output register routine
        extrn   Out_Pair:near           ; Output register pair routine
        extrn   Wait_Active:near        ; Wait for active display routine
        extrn   Logical_Width:word      ; Logical line width
        extrn   Unit_Horiz:word         ; Pixels/horizontal unit
        extrn   Unit_Vert:word          ; Scan lines/vertical unit
        extrn   Horiz_Size:word         ; Pixels/horizontal step  (Physical)
        extrn   Vert_Size:word          ; Scan lines/vertical step (Physical)
        extrn   Current_Horiz:word      ; Current horizontal pixel position
        extrn   Current_Vert:word       ; Current vertical scan line position
        extrn   Target_Horiz:word       ; Target horizontal pixel position
        extrn   Target_Vert:word        ; Target vertical scan line position
```

USING THE VERTICAL BLANKING INTERRUPT

```
Microsoft (R) Macro Assembler  Version 4.00                9/9/86 12:00:00

Vertical_INT    Sample Vertical Interrupt Routine          Page     1-2


                                      extrn   Max_Horiz:word         ; Maximum horizontal pixel position
                                      extrn   Max_Vert:word          ; Maximum vertical scan line position
                                      extrn   Horiz_Rate:word        ; Horizontal Pixels/update
                                      extrn   Vert_Rate:word         ; Vertical Scan lines/update
                                      extrn   Delay:word             ; Updates/Vertical frame
                                      extrn   Count:word             ; Current delay count value
                                      extrn   Pan_Table:word         ; Correct panning table address
                                      extrn   Start_Address:word     ; Start address save area
                                      extrn   Preset_Pan:word        ; Preset row scan/Pixel pan save area
                                      extrn   Scroll_Flag:byte       ; Scroll flag
                                      extrn   Vertical_End:byte      ; Vertical end register
                              ;
                              ;  LOCAL Equates
                              ;
 = 0000                       ABS_ZERO        equ     00h            ; Absolute zero segment (Int. vectors)
 = 0040                       BIOS_SEG        equ     40h            ; ROM BIOS data area segment
 = 0028                       INT_2_OFF       equ     Word Ptr [028h] ; Interrupt two offset address
 = 002A                       INT_2_SEG       equ     Word Ptr [02Ah] ; Interrupt two segment address
 = 0006                       STAT_PORT       equ     06h            ; Offset to input status port one
 = 0008                       PRESET_ROW      equ     08h            ; CRT controller preset row scan reg.
 = 000C                       START_HIGH      equ     0Ch            ; CRT controller start address MSW reg.
 = 000D                       START_LOW       equ     0Dh            ; CRT controller start address LSW reg.
 = 0011                       VERT_END        equ     11h            ; CRT controller vertical end register
 = 0010                       ENABLE_INT      equ     10h            ; Enable vertical interrupt bit
 = 03C2                       STATUS_REG      equ     3C2h           ; Input status register zero address
 = 0080                       CRT_INT         equ     80h            ; CRT interrupt flag bit
```

```
Microsoft (R) Macro Assembler  Version 4.00             9/9/86 12:00:00

Vertical_INT    Sample Vertical Interrupt Routine         Page     1-3


 = 03C0                         ATTR_INDEX      equ     3C0h            ; Attribute controller index port
 = 0013                         PIXEL_PAN       equ     13h             ; Attribute controller pixel pan reg.
 = 0020                         PALETTE         equ     20h             ; Attribute controller palette source
 = 00FF                         SCROLL_NEEDED   equ     0FFh            ; Scroll needed flag value
 = 0001                         SCROLLING       equ     01h             ; Scrolling flag value
 = 0020                         EOI             equ     20h             ; Interrupt controller EOI command
 = 0020                         CONTROL         equ     20h             ; Interrupt controller control port
 = 0063                         v_base          equ     word ptr 063h   ; CRT controller base address offset
                                ;
                                ;  Define the code segment
                                ;
 0000                           code    segment byte public 'CODE'      ; Standard code segment
                                        assume  cs:code, ds:nothing
```

```
Microsoft (R) Macro Assembler  Version 4.00            9/9/86 12:00:00

Vertical_INT    Sample Vertical Interrupt Routine       Page     1-4


                                    page
                                ;**************************************************************************
                                ;
                                ;       Perform_Scroll()
                                ;
                                ;       Registers on Entry:
                                ;
                                ;               None
                                ;
                                ;               The following variables MUST be set up on entry
                                ;
                                ;                       Target_Horiz  - Target horizontal pixel
                                ;                       Target_Vert   - Target vertical scan line
                                ;                       Scroll_Flag   - Scroll flag  (Scrolling or Needed)
                                ;                       Horiz_Rate    - Pixels/Update
                                ;                       Horiz_Size    - Pixels/Horizontal step   (Physical)
                                ;                       Vert_Rate     - Scan lines/Update
                                ;                       Vert_Size     - Scan lines/Vertical step (Physical)
                                ;                       Delay         - Vertical Frames/Update
                                ;
                                ;       Registers on Exit:
                                ;
                                ;               None
                                ;
                                ;**************************************************************************
 0000                           Perform_Scroll  Proc    Near            ; Perform the scroll procedure
```

```
Microsoft (R) Macro Assembler  Version 4.00             9/9/86 12:00:00

Vertical_INT    Sample Vertical Interrupt Routine        Page     1-5


0000  50                      push    ax                     ; Save the AX register value
0001  52                      push    dx                     ; Save the DX register value
0002  1E                      push    ds                     ; Save the DS register value
0003                      Scroll_Loop:
0003  FA                      cli                            ;;; Disable interrupts (Lockout)
0004  2E: A0 0000 E           mov     al,cs:[Scroll_Flag]    ;;; Get the current scroll flag
0008  FE C0                   inc     al                     ;;; Check for scrolling needed
000A  75 37                   jnz     No_Scroll              ;;; Jump if scrolling NOT needed
000C  FB                      sti                            ;;; Enable interrupts (Unlock)
000D  B8 0000                 mov     ax,ABS_ZERO            ; Get absolute zero segment address
0010  8E D8                   mov     ds,ax                  ; Setup to access the interrupt vectors
0012  A1 0028                 mov     ax,ds:[INT_2_OFF]      ; Get the current interrupt 2 offset
0015  2E: A3 01B9 R           mov     cs:[Old_Offset],ax     ; Save as the old vector offset
0019  A1 002A                 mov     ax,ds:[INT_2_SEG]      ; Get the current interrupt 2 segment
001C  2E: A3 01BB R           mov     cs:[Old_Segment],ax    ; Save as the old vector segment
0020  FA                      cli                            ;;; Disable interrupts (Lockout)
0021  8D 06 0048 R            lea     ax,Vert_INT            ;;; Get address of interrupt routine
0025  A3 0028                 mov     ds:[INT_2_OFF],ax      ;;; Setup the new interrupt 2 offset
0028  8C 0E 002A              mov     ds:[INT_2_SEG],cs      ;;; Setup the new interrupt 2 segment
002C  B8 0040                 mov     ax,BIOS_SEG            ;;; Get the BIOS data area segment
002F  8E D8                   mov     ds,ax                  ;;; Setup to access BIOS data segment
0031  8B 16 0063              mov     dx,ds:[v_base]         ;;; Get the CRT base port address
0035  B4 11                   mov     ah,VERT_END            ;;; Get the vertical end register index
0037  2E: A0 0000 E           mov     al,cs:[Vertical_End]   ;;; Get the vertical end register value
003B  E8 0000 E               call    Out_Pair               ;;; Clear the vertical interrupt
003E  0C 10                   or      al,ENABLE_INT          ;;; Setup to enable vertical interrupts
```

```
Microsoft (R) Macro Assembler  Version 4.00            9/9/86 12:00:00

Vertical_INT    Sample Vertical Interrupt Routine      Page     1-6


 0040  E8 0000 E                call    Out_Pair             ;;; Enable and start vertical interrupt
 0043                      No_Scroll:
 0043  FB                       sti                          ;;; Enable interrupts (Unlock)
 0044  1F                       pop     ds                   ; Restore the DS register value
 0045  5A                       pop     dx                   ; Restore the DX register value
 0046  58                       pop     ax                   ; Restore the AX register value
 0047  C3                       ret                          ; Return to the caller
                           Perform_Scroll  Endp              ; End of the perform scroll procedure
```

```
Microsoft (R) Macro Assembler  Version 4.00            9/9/86 12:00:00

Vertical_INT    Sample Vertical Interrupt Routine          Page     1-7


                                      page
                               ;*****************************************************************************
                               ;
                               ;       Vert_INT()
                               ;
                               ;       Registers on Entry:
                               ;
                               ;               None
                               ;
                               ;               The following variables MUST be set up on entry
                               ;
                               ;                       Target_Horiz  - Target horizontal pixel
                               ;                       Target_Vert   - Target vertical scan line
                               ;                       Scroll_Flag   - Scroll flag  (Scrolling or Needed)
                               ;                       Horiz_Rate    - Pixels/Update
                               ;                       Horiz_Size    - Pixels/Horizontal step   (Physical)
                               ;                       Vert_Rate     - Scan lines/Update
                               ;                       Vert_Size     - Scan lines/Vertical step (Physical)
                               ;                       Delay         - Vertical Frames/Update
                               ;
                               ;       Registers on Exit:
                               ;
                               ;               None
                               ;
                               ;*****************************************************************************
0048                           Vert_INT        Proc                    ; Vertical retrace interrupt procedure
```

```
Microsoft (R) Macro Assembler  Version 4.00              9/9/86 12:00:00

Vertical_INT    Sample Vertical Interrupt Routine        Page     1-8


 0048  50                                push   ax                   ;;; Save the AX register value
 0049  53                                push   bx                   ;;; Save the BX register value
 004A  51                                push   cx                   ;;; Save the CX register value
 004B  52                                push   dx                   ;;; Save the DX register value
 004C  1E                                push   ds                   ;;; Save the DS register value
 004D  BA 03C2                           mov    dx,STATUS_REG        ;;; Get the status port zero address
 0050  EC                                in     al,dx                ;;; Input the status value
 0051  A8 80                             test   al,CRT_INT           ;;; Check for CRT vertical interrupt
 0053  75 0A                             jnz    Truly                ;;; Jump if truly a CRT interrupt
 0055  1F                                pop    ds                   ;;; Restore the DS register value
 0056  5A                                pop    dx                   ;;; Restore the DX register value
 0057  59                                pop    cx                   ;;; Restore the CX register value
 0058  5B                                pop    bx                   ;;; Restore the BX register value
 0059  58                                pop    ax                   ;;; Restore the AX register value
 005A  2E: FF 2E 01B9 R                  jmp    cs:[Old_Vector]      ;;; Jump to original interrupt vector
 005F                          Truly:
 005F  B8 0040                           mov    ax,BIOS_SEG          ;;; Get the BIOS data area segment
 0062  8E D8                             mov    ds,ax                ;;; Setup to access BIOS data segment
 0064  2E: A0 0000 E                     mov    al,cs:[Scroll_Flag]  ;;; Get the current scroll flag
 0068  0A C0                             or     al,al                ;;; Check for scrolling needed
 006A  74 04                             jz     Do_Pan               ;;; Jump if last past through
 006C  FE C0                             inc    al                   ;;; Check for scrolling just started
 006E  74 2B                             jz     Setup_Next           ;;; Jump if scrolling just started
 0070                          Do_Pan:
 0070  2E: 8B 0E 0000 E                  mov    cx,cs:[Preset_Pan]   ;;; Get new preset and panning values
 0075  8B 16 0063                        mov    dx,ds:[v_base]       ;;; Get the CRT base port address
```

```
Microsoft (R) Macro Assembler  Version 4.00              9/9/86 12:00:00

Vertical_INT    Sample Vertical Interrupt Routine        Page     1-9


 0079  83 C2 06                   add    dx,STAT_PORT          ;;; Increment back to the status port
 007C  E8 0000 E                  call   Input                 ;;; Reset the attribute flip-flop
 007F  BA 03C0                    mov    dx,ATTR_INDEX         ;;; Get the attribute controller port
 0082  B4 33                      mov    ah,PIXEL_PAN+PALETTE  ;;; Get the pixel pan register index
 0084  8A C1                      mov    al,cl                 ;;; Get the new pixel panning value
 0086  E8 0000 E                  call   Out_Pair              ;;; Output the new pixel panning value
 0089  2E: FF 0E 0000 E           dec    Word Ptr cs:[Count]   ;;; Decrement the current delay count
 008E  74 03                      jz     Do_Update             ;;; Jump if time to do an update
 0090  E9 019C R                  jmp    Vert_Done             ;;; Go exit and wait for next interrupt
 0093                        Do_Update:
 0093  2E: A1 0000 E              mov    ax,cs:[Delay]         ;;; Get the desired delay count
 0097  2E: A3 0000 E              mov    cs:[Count],ax         ;;; Reset the current count value
 009B                        Setup_Next:
 009B  FB                         sti                          ;;; Reenable interrupts (Unlock)
 009C  B0 20                      mov    al,EOI                ; Get interrupt controller EOI command
 009E  E6 20                      out    CONTROL,al            ; Output EOI to interrupt control port
 00A0  2E: A0 0000 E              mov    al,cs:[Scroll_Flag]   ; Get the current scroll flag
 00A4  0A C0                      or     al,al                 ; Check for scrolling needed
 00A6  75 18                      jnz    Reset_Flag            ; Jump if more scrolling needed
 00A8                        Scroll_Done:
 00A8  B8 0000                    mov    ax,ABS_ZERO           ; Get the absolute zero segment value
 00AB  8E D8                      mov    ds,ax                 ; Setup to modify interrupt vector
 00AD  FA                         cli                          ;;; Disable interrupts (Lockout)
 00AE  2E: A1 01B9 R              mov    ax,cs:[Old_Offset]    ;;; Get the original INT 2 offset
 00B2  A3 0028                    mov    ds:[INT_2_OFF],ax     ;;; Restore the original INT 2 offset
 00B5  2E: A1 01BB R              mov    ax,cs:[Old_Segment]   ;;; Get the original INT 2 segment
```

```
Microsoft (R) Macro Assembler  Version 4.00              9/9/86 12:00:00

Vertical_INT    Sample Vertical Interrupt Routine        Page     1-10


 00B9  A3 002A                      mov   ds:[INT_2_SEG],ax      ;;; Restore the original INT 2 segment
 00BC  FB                           sti                          ;;; Enable interrupts (Unlock)
 00BD  E9 01B3 R                    jmp   INT_Return             ; Go return from the interrupt
 00C0                         Reset_Flag:
 00C0  32 C0                        xor   al,al                  ; Setup to reset the scroll flag
 00C2  2E: A2 0000 E                mov   cs:[Scroll_Flag],al    ; Reset the scroll flag value
 00C6  2E: A1 0000 E                mov   ax,cs:[Current_Vert]   ; Get the current vertical scan line
 00CA  2E: 3B 06 0000 E             cmp   ax,cs:[Target_Vert]    ; Compare against the target scan line
 00CF  74 34                        je    Do_Horiz               ; Jump if at the target value
 00D1  72 18                        jb    Scroll_Down            ; Jump if below target value
 00D3                         Scroll_Up:
 00D3  2E: C6 06 0000 E 01          mov   Byte Ptr cs:[Scroll_Flag],SCROLLING
 00D9  2E: 2B 06 0000 E             sub   ax,cs:[Vert_Rate]      ; Subtract the vertical rate value
 00DE  2E: 3B 06 0000 E             cmp   ax,cs:[Target_Vert]    ; Compare against target value again
 00E3  7D 1C                        jge   Set_Vert               ; Jump if not past the target value
 00E5  2E: A1 0000 E                mov   ax,cs:[Target_Vert]    ; Past target, use target value instead
 00E9  EB 16                        jmp   SHORT Set_Vert         ; Go save the new vertical scan line
 00EB                         Scroll_Down:
 00EB  2E: C6 06 0000 E 01          mov   Byte Ptr cs:[Scroll_Flag],SCROLLING
 00F1  2E: 03 06 0000 E             add   ax,cs:[Vert_Rate]      ; Add the vertical rate value
 00F6  2E: 3B 06 0000 E             cmp   ax,cs:[Target_Vert]    ; Compare against target value again
 00FB  7E 04                        jle   Set_Vert               ; Jump if not past the target value
 00FD  2E: A1 0000 E                mov   ax,cs:[Target_Vert]    ; Past target, use target value instead
 0101                         Set_Vert:
 0101  2E: A3 0000 E                mov   cs:[Current_Vert],ax   ; Update the current vertical scan line
 0105                         Do_Horiz:
```

```
Microsoft (R) Macro Assembler  Version 4.00              9/9/86 12:00:00

Vertical_INT    Sample Vertical Interrupt Routine         Page     1-11


 0105  2E: A1 0000 E                    mov     ax,cs:[Current_Horiz]   ; Get the current horizontal pixel
 0109  2E: 3B 06 0000 E                 cmp     ax,cs:[Target_Horiz]    ; Compare against the target pixel
 010E  74 34                            je      Do_Calculate            ; Jump if at the target value
 0110  72 18                            jb      Scroll_Right            ; Jump if below the target value
 0112                           Scroll_Left:
 0112  2E: C6 06 0000 E 01              mov     Byte Ptr cs:[Scroll_Flag],SCROLLING
 0118  2E: 2B 06 0000 E                 sub     ax,cs:[Horiz_Rate]      ; Subtract the horizontal rate value
 011D  2E: 3B 06 0000 E                 cmp     ax,cs:[Target_Horiz]    ; Compare against target value again
 0122  7D 1C                            jge     Set_Horiz               ; Jump if not past the target value
 0124  2E: A1 0000 E                    mov     ax,cs:[Target_Horiz]    ; Past target, use target value instead
 0128  EB 16                            jmp     SHORT Set_Horiz         ; Go save the new horizontal pixel
 012A                           Scroll_Right:
 012A  2E: C6 06 0000 E 01              mov     Byte Ptr cs:[Scroll_Flag],SCROLLING
 0130  2E: 03 06 0000 E                 add     ax,cs:[Horiz_Rate]      ; Add the horizontal rate value
 0135  2E: 3B 06 0000 E                 cmp     ax,cs:[Target_Horiz]    ; Compare against target value again
 013A  7E 04                            jle     Set_Horiz               ; Jump if not past the target value
 013C  2E: A1 0000 E                    mov     ax,cs:[Target_Horiz]    ; Past target, use target value instead
 0140                           Set_Horiz:
 0140  2E: A3 0000 E                    mov     cs:[Current_Horiz],ax   ; Update the current horizontal pixel
 0144                           Do_Calculate:
 0144  2E: A1 0000 E                    mov     ax,cs:[Current_Vert]    ; Get the current vertical scan line
 0148  99                               cwd                             ; Convert value to a double word
 0149  2E: F7 36 0000 E                 div     cs:[Vert_Size]          ; Divide by the vertical step size
 014E  8A EA                            mov     ch,dl                   ; Save preset row scan value (Remainder)
 0150  2E: F7 26 0000 E                 mul     cs:[Logical_Width]      ; Multiply by logical line length
 0155  8B D8                            mov     bx,ax                   ; Save result as new offset value
```

```
Microsoft (R) Macro Assembler  Version 4.00                9/9/86 12:00:00

Vertical_INT    Sample Vertical Interrupt Routine          Page     1-12


 0157  2E: A1 0000 E                 mov     ax,cs:[Current_Horiz]   ; Get the current horizontal pixel
 015B  99                            cwd                             ; Convert value to a double word
 015C  2E: F7 36 0000 E              div     cs:[Horiz_Size]         ; Divide by the horizontal step size
 0161  8B F2                         mov     si,dx                   ; Save the pixel shift in SI (Remainder)
 0163  2E: 8B 2E 0000 E              mov     bp,cs:[Pan_Table]       ; Get the correct panning table address
 0168  2E: 8A 0A                     mov     cl,cs:[bp+si]           ; Save the pixel panning value (Lookup)
 016B  03 D8                         add     bx,ax                   ; Add result to get new offset value
 016D  2E: 89 1E 0000 E              mov     cs:[Start_Address],bx   ; Save the new start address value
 0172  2E: 89 0E 0000 E              mov     cs:[Preset_Pan],cx      ; Save the new preset/pan values
 0177  8B 16 0063                    mov     dx,ds:[v_base]          ; Get the CRT base port address
 017B  83 C2 06                      add     dx,STAT_PORT            ; Increment to status port address
 017E  E8 0000 E                     call    wait_active             ; Wait till active display time
 0181  83 EA 06                      sub     dx,STAT_PORT            ; Decrement back to base port address
 0184  B4 0C                         mov     ah,START_HIGH           ; Get the start address high register
 0186  8A C7                         mov     al,bh                   ; Get MSB of the new start address
 0188  E8 0000 E                     call    Out_Pair                ; Output new start address high value
 018B  B4 0D                         mov     ah,START_LOW            ; Get the start address low register
 018D  8A C3                         mov     al,bl                   ; Get LSB of the new start address
 018F  E8 0000 E                     call    Out_Pair                ; Output new start address low value
 0192  B4 08                         mov     ah,PRESET_ROW           ; Get the preset row scan register
 0194  8A C5                         mov     al,ch                   ; Get the new preset row scan value
 0196  E8 0000 E                     call    Out_Pair                ; Output new preset row scan value
 0199  EB 06 90                      jmp     Vert_Exit               ; Go exit this vertical interrupt
 019C                        Vert_Done:
 019C  FB                            sti                             ;;; Reenable interrupts (Unlock)
 019D  B0 20                         mov     al,EOI                  ; Get interrupt controller EOI command
```

```
Microsoft (R) Macro Assembler  Version 4.00              9/9/86 12:00:00

Vertical_INT    Sample Vertical Interrupt Routine         Page     1-13


 019F  E6 20                       out     CONTROL,al            ; Output EOI to interrupt control port
 01A1                      Vert_Exit:
 01A1  8B 16 0063                  mov     dx,ds:[v_base]        ; Get the CRT base port address
 01A5  B4 11                       mov     ah,VERT_END           ; Get the vertical end register index
 01A7  2E: A0 0000 E               mov     al,cs:[Vertical_End]  ; Get the vertical end register value
 01AB  E8 0000 E                   call    Out_Pair              ; Clear the vertical interrupt
 01AE  0C 10                       or      al,ENABLE_INT         ; Setup to reenable the vertical int.
 01B0  E8 0000 E                   call    Out_Pair              ; Reenable the vertical interrupt
 01B3                      INT_Return:
 01B3  1F                          pop     ds                    ; Restore the DS register value
 01B4  5A                          pop     dx                    ; Restore the DX register value
 01B5  59                          pop     cx                    ; Restore the CX register value
 01B6  5B                          pop     bx                    ; Restore the BX register value
 01B7  58                          pop     ax                    ; Restore the AX register value
 01B8  CF                          iret                          ; Return to interrupted code
                           Vert_INT        Endp                  ; End of vertical interrupt procedure
                           ;
                           ;  Define the local data values
                           ;
 = 01B9                    Old_Vector      equ     This Dword    ; Original INT 2 vector save area
 01B9  0000                Old_Offset      dw      0             ; Original INT 2 vector offset
 01BB  0000                Old_Segment     dw      0             ; Original INT 2 vector segment
                           ;
                           ;  Define the end of the code segment
                           ;
 01BD                      code    ends
```

```
Microsoft (R) Macro Assembler  Version 4.00                 9/9/86 12:00:00

Vertical_INT    Sample Vertical Interrupt Routine           Page     1-14


                                   end                           ; End of the vertical interrupt module
```

B.7 PROGRAMMING THE COLOR PALETTE

This example illustrates how the color palette can be changed from the default color programming that was performed when the current BIOS mode was set.

```
Microsoft (R) Macro Assembler  Version 4.00              9/9/86 12:00:00

Color_Palette   Sample Color Palette Routine             Page     1-1


                                      page    30,120
                                      title   Color_Palette   Sample Color Palette Routine
                               ;***************************************************************************
                               ;
                               ;  Module Functional Description:
                               ;
                               ;            This module contains a routine to change the actual
                               ;      color of any of the current colors.
                               ;
                               ;***************************************************************************
                                      public  Change_Palette          ; Change color palette routine
                                      extrn   Input:near              ; Input register routine
                                      extrn   Output:near             ; Output register routine
                                      extrn   Out_Pair:near           ; Output register pair routine
                                      extrn   Wait_Vert:near          ; Wait for vertical retrace routine
                               ;
                               ;  LOCAL Equates
                               ;
 = 0040                        BIOS_SEG       equ     0040h           ; BIOS RAM data area segment (400h abs.)
 = 0010                        MAX_PALETTE    equ     10h             ; Maximum palette register number + 1
 = 0040                        MAX_COLOR      equ     40h             ; Maximum palette color value + 1
 = 0006                        STAT_PORT      equ     06h             ; Offset to input status port one
 = 03C0                        ATTR_INDEX     equ     3C0h            ; Attribute controller index port
 = 0020                        PALETTE        equ     20h             ; Attribute controller palette source
 = 0063                        v_base         equ     word ptr [63h]  ; CRT controller base port (BIOS Data)
                               ;
```

```
Microsoft (R) Macro Assembler  Version 4.00              9/9/86 12:00:00

Color_Palette   Sample Color Palette Routine             Page     1-2


                                ;  Define the code segment
                                ;
 0000                           code    segment byte public 'CODE'      ; Standard code segment
                                        assume  cs:code, ds:nothing
```

```
Microsoft (R) Macro Assembler  Version 4.00              9/9/86 12:00:00

Color_Palette   Sample Color Palette Routine             Page     1-3


                                   page
                            ;***************************************************************************
                            ;
                            ;       Change_Palette(Palette, Color)
                            ;
                            ;       Registers on Entry:
                            ;
                            ;               AH    - Palette number  (0..15)
                            ;               AL    - New color value (0..63)
                            ;
                            ;       Registers on Exit:
                            ;
                            ;               None
                            ;
                            ;               CY    - Set if invalid palette or color
                            ;
                            ;***************************************************************************
0000                        Change_Palette  Proc    Near            ; Change palette procedure
0000  50                            push    ax                      ; Save the AX register value
0001  52                            push    dx                      ; Save the DX register value
0002  1E                            push    ds                      ; Save the DS register value
0003  BA 0040                       mov     dx,BIOS_SEG             ; Get the BIOS data area segment value
0006  8E DA                         mov     ds,dx                   ; Set DS to the BIOS data area segment
0008  80 FC 10                      cmp     ah,MAX_PALETTE          ; Check the given palette number
000B  73 1D                         jae     Palette_Done            ; Jump if above the maximum value
000D  3C 40                         cmp     al,MAX_COLOR            ; Check the given color value
```

```
Microsoft (R) Macro Assembler  Version 4.00          9/9/86 12:00:00

Color_Palette   Sample Color Palette Routine         Page     1-4


 000F  73 19                      jae     Palette_Done          ; Jump if above the maximum value
 0011  8B 16 0063                 mov     dx,ds:[v_base]        ; Get CRT controller base port address
 0015  83 C2 06                   add     dx,STAT_PORT          ; Increment to the status port address
 0018  50                         push    ax                    ; Save the palette number and color
 0019  E8 0000 E                  call    Wait_Vert             ;;; Wait for vertical retrace period
 001C  58                         pop     ax                    ;;; Restore palette number and color
 001D  BA 03C0                    mov     dx,ATTR_INDEX         ;;; Get the attribute controller index
 0020  E8 0000 E                  call    Out_Pair              ;;; Output the new color value
 0023  B0 20                      mov     al,PALETTE            ;;; Setup to reenable attribute access
 0025  E8 0000 E                  call    Output                ;;; Reenable attribute palette access
 0028  FB                         sti                           ;;; Enable interrupts again (Unlock)
 0029  F9                         stc                           ; Set carry to indicate no errors
 002A                     Palette_Done:
 002A  F5                         cmc                           ; Complement carry to correct state
 002B  1F                         pop     ds                    ; Restore the DS register value
 002C  5A                         pop     dx                    ; Restore the DX register value
 002D  58                         pop     ax                    ; Restore the AX register value
 002E  C3                         ret                           ; Return to the caller
                          Change_Palette  Endp                  ; End of the change palette procedure
                          ;
                          ;  Define the end of the code segment
                          ;
 002F                     code    ends
                                  end                           ; End of the change palette module
```

B.8 MODIFYING VIDEO PARAMETERS

This example shows how a user-supplied video parameter table can be substituted for the default one located in the video BIOS ROM. In practice, the ROM-based parameter table is copied into an area of RAM and portions of subtables are modified. New BIOS modes can also be defined, starting with mode 17h and appending the respective subtable at offset +5C0h. Before changing any of the default parameters in these registers, be sure that the hardware environment is understood (for example, which monitors are connected to the video display controller and the timing that each monitor requires) for proper operation.

```
Microsoft (R) Macro Assembler  Version 4.00                 9/9/86 12:00:00

Modify          Sample Modify Parameters Routine            Page     1-1


                                        page    30,120
                                        title   Modify          Sample Modify Parameters Routine
                                ;*******************************************************************************
                                ;
                                ;  Module Functional Description:
                                ;
                                ;               This module contains a routine that copies the video parameter
                                ;       table into RAM so that it may be modified, it also contains a routine
                                ;       to return the address of a particular video mode's sub-table.
                                ;
                                ;*******************************************************************************
                                        public  Copy_Table              ; Copy video table routine
                                        public  Get_Address             ; Get video mode sub-table address
                                ;
                                ;  LOCAL Equates
                                ;
 = 0040                         BIOS_SEG        equ     0040h           ; BIOS RAM data area segment (400h abs.)
 = 001C                         SAVE_SIZE       equ     07h * 4         ; Save area table size (7 dword ptr's)
 = 0040                         ENTRY_SIZE      equ     40h             ; Video mode sub-table size (64 Bytes)
 = 05C0                         PARM_SIZE       equ     17h * ENTRY_SIZE; Video parameter table size (23 Modes)
 =                              V_OFFSET        equ     SAVE_SIZE       ; Video parameter table save offset
 = 0000                         V_PARM          equ     00h             ; Video parameter table pointer offset
 = 0000                         PARM_OFF        equ     00h             ; Video parameter table offset storage
 = 0002                         PARM_SEG        equ     02h             ; Video parameter table segment storage
 = 00A8                         v_off           equ     word ptr 0A8h   ; Video save area table offset storage
 = 00AA                         v_seg           equ     word ptr 0AAh   ; Video save area table segment storage
```

```
Microsoft (R) Macro Assembler  Version 4.00              9/9/86 12:00:00

Modify         Sample Modify Parameters Routine          Page     1-2


 = 00A8                        v_save          equ     dword ptr 0A8h  ; Video save area table pointer
                               ;
                               ;  Define the code segment
                               ;
 0000                          code    segment byte public 'CODE'      ; Standard code segment
                                       assume  cs:code, ds:nothing
```

```
Microsoft (R) Macro Assembler  Version 4.00          9/9/86 12:00:00

Modify          Sample Modify Parameters Routine         Page     1-3


                                         page
                                        ;*****************************************************************************
                                        ;
                                        ;       Copy_Table(Area)
                                        ;
                                        ;       Registers on Entry:
                                        ;
                                        ;               ES:0  - Pointer to save area (MUST be at least 1500 Bytes)
                                        ;
                                        ;       Registers on Exit:
                                        ;
                                        ;               None
                                        ;
                                        ;*****************************************************************************
 0000                                   Copy_Table      Proc    Near            ; Copy video parameters procedure
 0000  51                                       push    cx                      ; Save the CX register value
 0001  56                                       push    si                      ; Save the SI register value
 0002  57                                       push    di                      ; Save the DI register value
 0003  1E                                       push    ds                      ; Save the DS register value
 0004  B9 0040                                  mov     cx,BIOS_SEG             ; Get the BIOS data segment value
 0007  8E D9                                    mov     ds,cx                   ; Setup to access the BIOS data segment
 0009  C5 36 00A8                               lds     si,ds:[v_save]          ; Get the save area table pointer
 000D  33 FF                                    xor     di,di                   ; Setup to move the save area table
 000F  B9 001C                                  mov     cx,SAVE_SIZE            ; Get the save area table size
 0012  F3/ A4                                   rep     movsb                   ; Move the save area table in RAM
 0014  B9 0040                                  mov     cx,BIOS_SEG             ; Get the BIOS data segment value
```

```
Microsoft (R) Macro Assembler  Version 4.00                 9/9/86 12:00:00

Modify          Sample Modify Parameters Routine            Page     1-4


 0017  8E D9                             mov     ds,cx                     ; Setup to access the BIOS data segment
 0019  33 C9                             xor     cx,cx                     ; Setup to save the new save area addr.
 001B  89 0E 00A8                        mov     ds:[v_off],cx             ; Save the new save area offset value
 001F  8C 06 00AA                        mov     ds:[v_seg],es             ; Save the new save area segment value
 0023  26: C5 36 0000                    lds     si,dword ptr es:[V_PARM]; Get video parameter table address
 0028  BF 001C                           mov     di,offset V_OFFSET        ; Setup to move video parameter table
 002B  B9 05C0                           mov     cx,PARM_SIZE              ; Get the size of video parameter table
 002E  F3/ A4                            rep     movsb                     ; Move the video parameter table to RAM
 0030  26: C7 06 0000 001C               mov     word ptr es:[PARM_OFF],offset V_OFFSET
 0037  26: 8C 06 0002                    mov     es:[PARM_SEG],es          ; Save new video table segment value
 003C  1F                                pop     ds                        ; Restore the DS register value
 003D  5F                                pop     di                        ; Restore the DI register value
 003E  5E                                pop     si                        ; Restore the SI register value
 003F  59                                pop     cx                        ; Restore the CX register value
 0040  C3                                ret                               ; Return to the caller
                               Copy_Table        Endp                      ; End of the copy table procedure
```

```
Microsoft (R) Macro Assembler  Version 4.00             9/9/86 12:00:00

Modify          Sample Modify Parameters Routine        Page     1-5


                                        page
                                ;***********************************************************************
                                ;
                                ;       Get_Address(Mode)
                                ;
                                ;       Registers on Entry:
                                ;
                                ;               AL    - Desired mode value (MUST be actual number)
                                ;
                                ;       Registers on Exit:
                                ;
                                ;               ES:DI - Pointer to mode sub-table
                                ;
                                ;***********************************************************************
 0041                           Get_Address     Proc    Near            ; Get mode sub-table address procedure
 0041  50                               push    ax                      ; Save the AX register value
 0042  1E                               push    ds                      ; Save the DS register value
 0043  B4 40                            mov     ah,ENTRY_SIZE           ; Get mode sub-table entry size (Bytes)
 0045  F6 E4                            mul     ah                      ; Multiply mode value by entry size
 0047  BF 0040                          mov     di,BIOS_SEG             ; Get the BIOS data area segment
 004A  8E DF                            mov     ds,di                   ; Setup to access the BIOS data area
 004C  C4 3E 00A8                       les     di,ds:[v_save]          ; Get the save area table address
 0050  26: C4 3D                        les     di,dword ptr es:[di + V_PARM]
 0053  03 F8                            add     di,ax                   ; Adjust to desired mode sub-table
 0055  1F                               pop     ds                      ; Restore the DS register value
 0056  58                               pop     ax                      ; Restore the AX register value
```

MODIFYING VIDEO PARAMETERS

```
Microsoft (R) Macro Assembler  Version 4.00              9/9/86 12:00:00

Modify          Sample Modify Parameters Routine          Page     1-6


 0057  C3                                ret                        ; Return to the caller
                                 Get_Address     Endp               ; End of get address procedure
                                 ;
                                 ;  Define the end of the code segment
                                 ;
 0058                            code    ends
                                         end                        ; End of the modify module
```

B.9 MAKING SPECIAL FONTS RESIDENT

Programming example B.2 showed how a different character font could be loaded. However, by making the font resident, the new font will automatically be substituted every time a BIOS mode is set.

MAKING SPECIAL FONTS RESIDENT

```
Microsoft (R) Macro Assembler  Version 4.00                 9/9/86 12:00:00

Resident_Font   Sample Resident Font Routine                Page     1-1


                                      page    30,120
                                      title   Resident_Font   Sample Resident Font Routine
                                ;*****************************************************************************
                                ;
                                ;  Module Functional Description:
                                ;
                                ;               This module contains routines to setup resident graphics or
                                ;       text fonts, i.e. the font is loaded automatically on a set mode BIOS
                                ;       call.
                                ;
                                ;       Note -  For this routine to work, the video save area table MUST
                                ;               reside in RAM because these routines modify that table.
                                ;               The default values for this table are the ones in the video
                                ;               ROM; they are copied to RAM by the Copy_Table routine or a
                                ;               user program.
                                ;
                                ;*****************************************************************************
                                      public  Text_Font              ; Make a text font resident routine
                                      public  Graphics_Font          ; Make a graphics font resident routine
                                ;
                                ;  LOCAL Equates
                                ;
 = 0040                         BIOS_SEG        equ     0040h        ; BIOS RAM data area segment (400h abs.)
 = 0008                         V_TEXT          equ     08h          ; Auxiliary text font pointer offset
 = 0008                         V_TEXT_OFF      equ     08h          ; Auxiliary text font offset
 = 000A                         V_TEXT_SEG      equ     0Ah          ; Auxiliary text font segment
```

```
Microsoft (R) Macro Assembler  Version 4.00              9/9/86 12:00:00

Resident_Font   Sample Resident Font Routine             Page     1-2


 = 000C                        V_GRAPHIC       equ     0Ch              ; Auxiliary graphics font pointer offset
 = 000C                        V_GRAPHIC_OFF   equ     0Ch              ; Auxiliary graphics font offset
 = 000E                        V_GRAPHIC_SEG   equ     0Eh              ; Auxiliary graphics font segment
 = 00A8                        v_save          equ     dword ptr 0A8h  ; Video save area table pointer
                               ;
                               ;  Define the code segment
                               ;
 0000                          code    segment byte public 'CODE'      ; Standard code segment
                                       assume  cs:code, ds:nothing
```

MAKING SPECIAL FONTS RESIDENT

```
Microsoft (R) Macro Assembler  Version 4.00                 9/9/86 12:00:00

Resident_Font   Sample Resident Font Routine                Page     1-3


                                  page
                               ;*****************************************************************************
                               ;
                               ;      Text_Font(Font_Pointer)
                               ;
                               ;      Registers on Entry:
                               ;
                               ;              ES:DI - Pointer to auxiliary text font table
                               ;
                               ;      Registers on Exit:
                               ;
                               ;              None
                               ;
                               ;              CY    - Set if save table NOT in RAM
                               ;
                               ;*****************************************************************************
0000                           Text_Font       Proc    Near            ; Resident text mode font procedure
0000  56                               push    si                      ; Save the SI register value
0001  1E                               push    ds                      ; Save the DS register value
0002  BE 0040                          mov     si,BIOS_SEG             ; Get the BIOS data area segment value
0005  8E DE                            mov     ds,si                   ; Setup to access the BIOS data area
0007  C5 36 00A8                       lds     si,ds:[v_save]          ; Get the video save table address
000B  E8 0038 R                        call    Test_Memory             ; Check for table in RAM area
000E  72 09                            jc      Text_Return             ; Jump if table NOT in RAM area
0010  89 3E 0008                       mov     ds:[V_TEXT_OFF],di      ; Save the text font table offset
0014  8C 06 000A                       mov     ds:[V_TEXT_SEG],es      ; Save the text font table segment
```

```
Microsoft (R) Macro Assembler  Version 4.00             9/9/86 12:00:00

Resident_Font   Sample Resident Font Routine            Page     1-4


0018  F8                        clc                        ; Clear carry indicating no error
0019                    Text_Return:
0019  1F                        pop     ds                 ; Restore the DS register value
001A  5E                        pop     si                 ; Restore the SI register value
001B  C3                        ret                        ; Return to the caller
                        Text_Font       Endp               ; End of the text font procedure
```

```
Microsoft (R) Macro Assembler  Version 4.00                 9/9/86 12:00:00

Resident_Font   Sample Resident Font Routine                Page     1-5


                                        page
                                ;******************************************************************************
                                ;
                                ;       Graphics_Font(Font_Pointer)
                                ;
                                ;       Registers on Entry:
                                ;
                                ;               ES:DI - Pointer to auxiliary graphics font table
                                ;
                                ;       Registers on Exit:
                                ;
                                ;               None
                                ;
                                ;               CY    - Set if save table NOT in RAM
                                ;
                                ;******************************************************************************
 001C                           Graphics_Font   Proc    Near             ; Resident graphics mode font procedure
 001C  56                               push    si                       ; Save the SI register value
 001D  1E                               push    ds                       ; Save the DS register value
 001E  BE 0040                          mov     si,BIOS_SEG              ; Get the BIOS data area segment value
 0021  8E DE                            mov     ds,si                    ; Setup to access the BIOS data area
 0023  C5 36 00A8                       lds     si,ds:[v_save]           ; Get the video save table address
 0027  E8 0038 R                        call    Test_Memory              ; Check for table in RAM area
 002A  72 09                            jc      Graphics_Return          ; Jump if table NOT in RAM area
 002C  89 3E 000C                       mov     ds:[V_GRAPHIC_OFF],di    ; Save the graphics font table offset
 0030  8C 06 000E                       mov     ds:[V_GRAPHIC_SEG],es    ; Save the graphics font table segment
```

```
Microsoft (R) Macro Assembler  Version 4.00                 9/9/86 12:00:00

Resident_Font   Sample Resident Font Routine                Page     1-6


 0034  F8                         clc                            ; Clear carry indicating no error
 0035                     Graphics_Return:
 0035  1F                         pop     ds                     ; Restore the DS register value
 0036  5E                         pop     si                     ; Restore the SI register value
 0037  C3                         ret                            ; Return to the caller
                          Graphics_Font   Endp                   ; End of the graphics font procedure
```

MAKING SPECIAL FONTS RESIDENT

```
Microsoft (R) Macro Assembler  Version 4.00              9/9/86 12:00:00

Resident_Font   Sample Resident Font Routine             Page     1-7


                                     page
                             ;****************************************************************************
                             ;
                             ;       Test_Memory(Pointer)
                             ;
                             ;       Registers on Entry:
                             ;
                             ;               DS:SI - Pointer to memory byte to check
                             ;
                             ;       Registers on Exit:
                             ;
                             ;               None
                             ;
                             ;               CY    - Set if memory at DS:SI does NOT exist
                             ;               DF    - The direction flag is ALWAYS cleared
                             ;
                             ;****************************************************************************
 0038                        Test_Memory     Proc    Near            ; Test memory procedure
 0038  50                            push    ax                      ; Save the AX register value
 0039  8A 04                         mov     al,ds:[si]              ; Save the byte to be tested
 003B  32 E4                         xor     ah,ah                   ; Zero the AH register value
 003D  88 24                         mov     ds:[si],ah              ; Set the test location to a zero
 003F  EB 00                         jmp     SHORT Flush_Pipeline    ; Flush the CPU pipeline queue
 0041                        Flush_Pipeline:
 0041  FD                            std                             ; Fill              (STD = FDh)
 0042  FD                            std                             ;     the           (STD = FDh)
```

```
Microsoft (R) Macro Assembler  Version 4.00            9/9/86 12:00:00

Resident_Font   Sample Resident Font Routine            Page     1-8


 0043  FD                        std                          ;          bus         (STD = FDh)
 0044  FD                        std                          ;              with    (STD = FDh)
 0045  FD                        std                          ;                  ones (STD = FDh)
 0046  3A 24                     cmp     ah,ds:[si]           ; Check for zero at the test location
 0048  F8                        clc                          ; Clear carry in case memory present
 0049  74 01                     je      Test_Return          ; Jump if zero there (Memory present)
 004B  F9                        stc                          ; No memory here, set carry flag
 004C                    Test_Return:
 004C  88 04                     mov     ds:[si],al           ; Restore the test location from AL
 004E  FC                        cld                          ; Clear the direction flag (Set by STD)
 004F  58                        pop     ax                   ; Restore the AX register value
 0050  C3                        ret                          ; Return to the caller
                         Test_Memory     Endp                 ; End of test memory procedure
                         ;
                         ;  Define the end of the code segment
                         ;
 0051                    code    ends
                                 end                          ; End of the resident module
```

B.10 SWITCHING BETWEEN COLOR TEXT AND MONOCHROME TEXT DISPLAY MODES

When the COMPAQ Enhanced Color Graphics Board is the only video controller in the system and it is driving a COMPAQ Dual-Mode Monitor, it has the ability to be reconfigured by software to either the color or monochrome configurations. This example shows how to verify the conditions stated above and perform a switch between configurations.

```
Microsoft (R) Macro Assembler  Version 4.00              9/9/86 12:00:00

Switch_Mode     Sample Switch Mode Routine                Page     1-1


                                        page    30,120
                                        title   Switch_Mode     Sample Switch Mode Routine
                                ;*****************************************************************************
                                ;
                                ;  Module Functional Description:
                                ;
                                ;               This module contains routines to switch the COMPAQ
                                ;       enhanced color graphics board between color/graphics and
                                ;       monochrome emulation modes.
                                ;
                                ;*****************************************************************************
                                        public  Switch_Color            ; Switch to color emulation routine
                                        public  Switch_Mono             ; Switch to monochrome emulation routine
                                        extrn   Present:near            ; Check for presence routine
                                        extrn   Active:near             ; Check for adapter active routine
                                        extrn   Input:near              ; Input a register routine
                                        extrn   Output:near             ; Output a register routine
                                ;
                                ;  LOCAL Equates
                                ;
 = 0040                         BIOS_SEG        equ     0040h           ; BIOS RAM data area segment (400h abs.)
 = 0000                         SET_MODE        equ     00h             ; BIOS set video mode function
 = 0003                         COLOR_MODE      equ     03h             ; 80 Column color text mode value
 = 0007                         MONO_MODE       equ     07h             ; 80 Column monochrome mode value
 = 00BF                         EXT_FUNCTION    equ     0BFh            ; Extended video BIOS function code
 = 0003                         GET_ENV         equ     03h             ; Get video environment sub-function
```

```
Microsoft (R) Macro Assembler  Version 4.00              9/9/86 12:00:00

Switch_Mode     Sample Switch Mode Routine               Page     1-2


 = 0010                         VIDEO        equ     10h              ; BIOS video interrupt number
 = 0001                         DSM          equ     01h              ; Dual mode monitor type number
 = 03B4                         MONO_CRTC    equ     03B4h            ; Monochrome CRTC index port address
 = 03D4                         COLOR_CRTC   equ     03D4h            ; Color/graphics CRTC index address
 = 000E                         CURSOR_HIGH  equ     0Eh              ; Cursor position high register index
 = 00A5                         PATTERN      equ     0A5h             ; Register present test pattern
 = 0010                         v_config     equ     byte ptr 10h     ; First byte of configuration word
 = 0020                         COLOR        equ     20h              ; 80 Column color configuration
 = 0030                         MONO         equ     30h              ; 80 Column monochrome configuration
 = 0087                         v_info       equ     byte ptr 87h     ; Miscellaneous video information
 = 0002                         MONO_DISPLAY equ     02h              ; Monochrome display attached bit
                                ;
                                ;  Define the code segment
                                ;
 0000                           code    segment byte public 'CODE'    ; Standard code segment
                                        assume  cs:code, ds:nothing
```

```
Microsoft (R) Macro Assembler  Version 4.00            9/9/86 12:00:00

Switch_Mode     Sample Switch Mode Routine              Page     1-3


                                 page
                              ;*************************************************************************
                              ;
                              ;       Switch_Color()
                              ;
                              ;       Registers on Entry:
                              ;
                              ;               None
                              ;
                              ;       Registers on Exit:
                              ;
                              ;               None
                              ;               CY    - Set if cannot change emulation modes
                              ;
                              ;*************************************************************************
 0000                         Switch_Color    Proc    Near            ; Switch to color mode procedure
 0000  50                             push    ax                      ; Save the AX register value
 0001  1E                             push    ds                      ; Save the DS register value
 0002  E8 0000 E                      call    Present                 ; Check for enhanced adapter present
 0005  72 30                          jc      Color_Return            ; Jump if NO enhanced adapter present
 0007  F9                             stc                             ; Set carry in case NOT COMPAQ adapter
 0008  74 2D                          jz      Color_Return            ; Jump if NON-COMPAQ enhanced adapter
 000A  E8 0000 E                      call    Active                  ; Call to check for adapter active
 000D  72 28                          jc      Color_Return            ; Jump if this adapter NOT active
 000F  E8 006F R                      call    Monitor                 ; Call routine to check monitor type
 0012  72 23                          jc      Color_Return            ; Jump if NOT a dual mode monitor
```

SWITCHING BETWEEN COLOR TEXT AND MONOCHROME TEXT DISPLAY MODES

```
Microsoft (R) Macro Assembler  Version 4.00              9/9/86 12:00:00

Switch_Mode     Sample Switch Mode Routine               Page     1-4


 0014  BA 03D4                      mov    dx,COLOR_CRTC          ; Get the color/graphics CRTC address
 0017  E8 008E R                    call   Check                  ; Call routine to check for adapter
 001A  72 1B                        jc     Color_Return           ; Jump if color/graphics already present
 001C  B8 0040                      mov    ax,BIOS_SEG            ; Get the BIOS data area segment value
 001F  8E D8                        mov    ds,ax                  ; Setup to access the BIOS data area
 0021  80 26 0087 FD                and    ds:[v_info],not MONO_DISPLAY
 0026  80 26 0010 CF                and    ds:[v_config],not MONO  ; Strip the video configuration bits
 002B  80 0E 0010 20                or     ds:[v_config],COLOR    ; Set configuration to color/graphics
 0030  B4 00                        mov    ah,SET_MODE            ; Get the BIOS set mode function
 0032  B0 03                        mov    al,COLOR_MODE          ; Get the color text mode value
 0034  CD 10                        int    VIDEO                  ; Set the mode to 80 column color
 0036  F8                           clc                           ; Clear carry indicating no errors
 0037                          Color_Return:
 0037  1F                           pop    ds                     ; Restore the DS register value
 0038  58                           pop    ax                     ; Restore the AX register value
 0039  C3                           ret                           ; Return to the caller
                               Switch_Color  Endp                 ; End of the color mode procedure
```

```
Microsoft (R) Macro Assembler  Version 4.00                  9/9/86 12:00:00

Switch_Mode     Sample Switch Mode Routine                   Page     1-5


                                        page
                                ;*******************************************************************************
                                ;
                                ;       Switch_Mono()
                                ;
                                ;       Registers on Entry:
                                ;
                                ;               None
                                ;
                                ;       Registers on Exit:
                                ;
                                ;               None
                                ;               CY    - Set if cannot change emulation modes
                                ;
                                ;*******************************************************************************
003A                            Switch_Mono     Proc    Near             ; Switch to monochrome mode procedure
003A  50                                push    ax                       ; Save the AX register value
003B  1E                                push    ds                       ; Save the DS register value
003C  E8 0000 E                         call    Present                  ; Check for enhanced adapter present
003F  72 2B                             jc      Mono_Return              ; Jump if NO enhanced adapter present
0041  F9                                stc                              ; Set carry in case NOT COMPAQ adapter
0042  74 28                             jz      Mono_Return              ; Jump if NON-COMPAQ enhanced adapter
0044  E8 0000 E                         call    Active                   ; Call to check for adapter active
0047  72 23                             jc      Mono_Return              ; Jump if this adapter NOT active
0049  E8 006F R                         call    Monitor                  ; Call routine to check monitor type
004C  72 1E                             jc      Mono_Return              ; Jump if NOT a dual mode monitor
```

SWITCHING BETWEEN COLOR TEXT AND MONOCHROME TEXT DISPLAY MODES

```
Microsoft (R) Macro Assembler  Version 4.00                 9/9/86 12:00:00

Switch_Mode     Sample Switch Mode Routine                  Page     1-6


 004E  BA 03B4                          mov     dx,MONO_CRTC            ; Get the monochrome CRTC address
 0051  E8 008E R                        call    Check                   ; Call routine to check for adapter
 0054  72 16                            jc      Mono_Return             ; Jump if monochrome already present
 0056  B8 0040                          mov     ax,BIOS_SEG             ; Get the BIOS data area segment value
 0059  8E D8                            mov     ds,ax                   ; Setup to access the BIOS data area
 005B  80 0E 0087 02                    or      ds:[v_info],MONO_DISPLAY; Set monochrome display attached bit
 0060  80 0E 0010 30                    or      ds:[v_config],MONO      ; Set configuration to monochrome
 0065  B4 00                            mov     ah,SET_MODE             ; Get the BIOS set mode function
 0067  B0 07                            mov     al,MONO_MODE            ; Get the monochrome mode value
 0069  CD 10                            int     VIDEO                   ; Set the mode to 80 column monochrome
 006B  F8                               clc                             ; Clear carry indicating no errors
 006C                          Mono_Return:
 006C  1F                               pop     ds                      ; Restore the DS register value
 006D  58                               pop     ax                      ; Restore the AX register value
 006E  C3                               ret                             ; Return to the caller
                               Switch_Mono      Endp                    ; End of the monochrome mode procedure
```

```
Microsoft (R) Macro Assembler  Version 4.00             9/9/86 12:00:00

Switch_Mode     Sample Switch Mode Routine              Page     1-7


                                page
                          ;***************************************************************************
                          ;
                          ;     Monitor()
                          ;
                          ;     Registers on Entry:
                          ;
                          ;             None
                          ;
                          ;     Registers on Exit:
                          ;
                          ;             None
                          ;             CY    - Set if NOT a dual mode monitor present
                          ;
                          ;***************************************************************************
006F                      Monitor       Proc    Near            ; Check monitor type procedure
006F  50                        push    ax                      ; Save the AX register value
0070  53                        push    bx                      ; Save the BX register value
0071  51                        push    cx                      ; Save the CX register value
0072  52                        push    dx                      ; Save the DX register value
0073  B4 BF                     mov     ah,EXT_FUNCTION         ; Get the extended BIOS function code
0075  B0 03                     mov     al,GET_ENV              ; Get the get environment sub-function
0077  33 DB                     xor     bx,bx                   ; Zero the BX register value
0079  CD 10                     int     VIDEO                   ; Get the video environment
007B  8A C2                     mov     al,dl                   ; Get the external monitor type
007D  0A FF                     or      bh,bh                   ; Check for external as active monitor
```

```
Microsoft (R) Macro Assembler  Version 4.00                9/9/86 12:00:00

Switch_Mode     Sample Switch Mode Routine                 Page     1-8


 007F  74 02                    jz      Check_Type          ; Jump if external monitor is active
 0081  8A C6                    mov     al,dh               ; Get the internal monitor type
 0083                   Check_Type:
 0083  3C 01                    cmp     al,DSM              ; Check for a dual mode monitor
 0085  F9                       stc                         ; Set carry in case NOT a dual mode
 0086  75 01                    jne     All_Done            ; Jump if NOT a dual mode monitor
 0088  F8                       clc                         ; Clear carry indicating dual mode
 0089                   All_Done:
 0089  5A                       pop     dx                  ; Restore the DX register value
 008A  59                       pop     cx                  ; Restore the CX register value
 008B  5B                       pop     bx                  ; Restore the BX register value
 008C  58                       pop     ax                  ; Restore the AX register value
 008D  C3                       ret                         ; Return to the caller
                        Monitor         Endp                ; End of monitor type procedure
```

```
Microsoft (R) Macro Assembler  Version 4.00            9/9/86 12:00:00

Switch_Mode     Sample Switch Mode Routine             Page     1-9


                                   page
                              ;***************************************************************************
                              ;
                              ;    Check(Address)
                              ;
                              ;    Registers on Entry:
                              ;
                              ;         DX    - CRTC address of adapter to check for
                              ;
                              ;    Registers on Exit:
                              ;
                              ;         None
                              ;         CY    - Set if adapter IS present
                              ;
                              ;***************************************************************************
008E                          Check          Proc    Near           ; Check for adapter present procedure
008E  50                           push      ax                     ; Save the AX register value
008F  53                           push      bx                     ; Save the BX register value
0090  B0 0E                        mov       al,CURSOR_HIGH         ; Get the cursor position high index
0092  FA                           cli                              ;;; Disable interrupts (Lockout)
0093  E8 0000 E                    call      Output                 ;;; Output cursor high index value
0096  42                           inc       dx                     ;;; Increment to CRTC data port address
0097  E8 0000 E                    call      Input                  ;;; Get the current cursor high value
009A  8A D8                        mov       bl,al                  ;;; Save the cursor high value in BL
009C  B0 A5                        mov       al,PATTERN             ;;; Get the register test pattern
009E  E8 0000 E                    call      Output                 ;;; Output the register test pattern
```

SWITCHING BETWEEN COLOR TEXT AND MONOCHROME TEXT DISPLAY MODES

```
Microsoft (R) Macro Assembler  Version 4.00              9/9/86 12:00:00

Switch_Mode     Sample Switch Mode Routine               Page     1-10


 00A1  E8 0000 E                   call    Input               ;;; Read the current register value
 00A4  8A F8                       mov     bh,al               ;;; Save the current value in BH
 00A6  8A C3                       mov     al,bl               ;;; Get the original value back into AL
 00A8  E8 0000 E                   call    Output              ;;; Restore the original register value
 00AB  FB                          sti                         ;;; Enable interrupts (Unlock)
 00AC  4A                          dec     dx                  ; Decrement back to index port address
 00AD  80 FF A5                    cmp     bh,PATTERN          ; Check for adapter as present
 00B0  F9                          stc                         ; Set carry in case adapter present
 00B1  74 01                       je      Check_Done          ; Jump if the adapter IS present
 00B3  F8                          clc                         ; Clear carry indicating NO adapter
 00B4                      Check_Done:
 00B4  5B                          pop     bx                  ; Restore the BX register value
 00B5  58                          pop     ax                  ; Restore the AX register value
 00B6  C3                          ret                         ; Return to the caller
                           Check           Endp                ; End of the check adapter procedure
                           ;
                           ;  Define the end of the code segment
                           ;
 00B7                      code    ends
                                   end                         ; End of the switch mode module
```

B.11 MISCELLANEOUS ROUTINES

This module contains several miscellaneous subroutines that are called by various routines in the preceeding sections.

Note: CALL and RET instructions, surrounding the OUT instructions, cause the instruction prefetch queue to be purged. This action introduces a minimum delay between I/O cycles, as required by many large-scale-integrated devices.

```
Microsoft (R) Macro Assembler  Version 4.00              9/9/86 12:00:00

Miscellaneous   Sample Programs Miscellaneous Routines      Page     1-1


                                page    30,120
                                title   Miscellaneous   Sample Programs Miscellaneous Routines
                        ;******************************************************************************
                        ;
                        ;  Module Functional Description:
                        ;
                        ;               This module contains the following miscellaneous routines
                        ;       used by the sample routines:
                        ;
                        ;               1) Present    - Detects enhanced adapter presence
                        ;
                        ;               2) Active     - Determines the active adapter
                        ;
                        ;               3) Actual     - Gets actual mode sub-table number
                        ;
                        ;               4) Output     - Outputs a register value
                        ;
                        ;               5) Out_Pair   - Outputs a register pair value
                        ;
                        ;               6) Input      - Inputs a register value
                        ;
                        ;               7) Wait_Vert  - Waits for vertical retrace
                        ;
                        ;               8) Wait_Active- Waits for active display time
                        ;
                        ;******************************************************************************
```

```
Microsoft (R) Macro Assembler  Version 4.00            9/9/86 12:00:00

Miscellaneous   Sample Programs Miscellaneous Routines     Page     1-2


                               public  Present              ; Check presence routine
                               public  Active               ; Check active routine
                               public  Actual               ; Get actual mode routine
                               public  Output               ; Output register value routine
                               public  Out_Pair             ; Output register pair routine
                               public  Input                ; Input register value routine
                               public  Wait_Vert            ; Wait for vertical retrace routine
                               public  Wait_Active          ; Wait for active display routine
                          ;
                          ;  LOCAL Equates
                          ;
 = 0040                   BIOS_SEG     equ    40h           ; ROM BIOS data area segment
 = 0010                   VIDEO        equ    10h           ; BIOS video interrupt number
 = 0012                   ALT_SELECT   equ    12h           ; Alternate select function code
 = 0010                   RET_INFO     equ    10h           ; Return EGA information sub-function
 = 00BF                   VIDEO_ENV    equ    0BFh          ; Video environment function code
 = 0003                   GET_ENV      equ    03h           ; Get current environment sub-function
 = 0008                   INACTIVE     equ    08h           ; Enhanced adapter active flag bit
 = 0003                   COL_MODE     equ    03h           ; Color/Graphics text mode (Maximum)
 = 0007                   MONO_MODE    equ    07h           ; Monochrome text mode
 = 000F                   SWITCH_MASK  equ    0Fh           ; Configuration switch mask value
 = 0009                   ENH_PRI      equ    09h           ; Primary enhanced switch value
 = 0003                   ENH_SEC      equ    03h           ; Secondary enhanced switch value
 = 0002                   GRAPH_ADJUST equ    002h          ; Hi-res graphics mode adjust value
 = 0013                   TEXT_ADJUST  equ    013h          ; Enhanced text mode adjust value
 = 000F                   HIGH_GRAPH   equ    0Fh           ; Hi-resolution graphics mode value
```

```
Microsoft (R) Macro Assembler  Version 4.00             9/9/86 12:00:00

Miscellaneous   Sample Programs Miscellaneous Routines      Page     1-3


 = 0010                       ENH_GRAPH       equ     10h             ; Enhanced graphics mode value
 = 0060                       EXT_MEM_PRE     equ     60h             ; Extended memory present test value
 = 0008                       VERT            equ     08h             ; Vertical retrace status bit
 = 0001                       ENABLE          equ     01h             ; Display enable status bit
 = 0087                       v_info          equ     BYTE PTR 87h    ; Video information byte (BIOS Data)
 = 0088                       v_config        equ     BYTE PTR 88h    ; Video configuration byte (BIOS Data)
                              ;
                              ;  Define the code segment
                              ;
 0000                         code    segment byte public 'CODE'      ; Standard code segment
                                      assume  cs:code, ds:nothing
```

```
Microsoft (R) Macro Assembler  Version 4.00               9/9/86 12:00:00

Miscellaneous   Sample Programs Miscellaneous Routines      Page     1-4


                                   page
                              ;*****************************************************************************
                              ;
                              ;     Present()
                              ;
                              ;     Registers on Entry:
                              ;
                              ;             None
                              ;
                              ;     Registers on Exit:
                              ;
                              ;             None
                              ;             CY    - Carry set if NO enhanced graphics board present
                              ;             ZR    - Zero set if a NON-COMPAQ enhanced graphics board
                              ;
                              ;*****************************************************************************
 0000                         Present        Proc    Near            ; Check presence procedure
 0000  50                           push     ax                      ; Save the AX register value
 0001  53                           push     bx                      ; Save the BX register value
 0002  51                           push     cx                      ; Save the CX register value
 0003  52                           push     dx                      ; Save the DX register value
 0004  B4 12                        mov      ah,ALT_SELECT           ; Setup the alternate select function
 0006  32 C0                        xor      al,al                   ; Zero the AL register
 0008  B3 10                        mov      bl,RET_INFO             ; Setup to return EGA information
 000A  CD 10                        int      VIDEO                   ; Get the EGA information
 000C  80 FB 10                     cmp      bl,RET_INFO             ; Check for EGA information returned
```

```
Microsoft (R) Macro Assembler  Version 4.00               9/9/86 12:00:00

Miscellaneous   Sample Programs Miscellaneous Routines      Page     1-5


 000F  F9                       stc                     ; Set carry flag in case NO EGA present
 0010  74 0D                    je      Return          ; Jump if no information was returned
 0012  B4 BF                    mov     ah,VIDEO_ENV    ; Setup for video environment function
 0014  B0 03                    mov     al,GET_ENV      ; Setup to get current video environment
 0016  33 DB                    xor     bx,bx           ; Zero the BX register
 0018  33 C9                    xor     cx,cx           ; Setup CX to a know state (Zero)
 001A  CD 10                    int     VIDEO           ; Get the current video environment
 001C  0A C9                    or      cl,cl           ; Check for information returned
 001E  F8                       clc                     ; Clear carry (Enhanced board present)
 001F                 Return:
 001F  5A                       pop     dx              ; Restore the DX register value
 0020  59                       pop     cx              ; Restore the CX register value
 0021  5B                       pop     bx              ; Restore the BX register value
 0022  58                       pop     ax              ; Restore the AX register value
 0023  C3                       ret                     ; Return to the caller
                      Present           Endp            ; End of the check presence procedure
```

```
Microsoft (R) Macro Assembler  Version 4.00              9/9/86 12:00:00

Miscellaneous   Sample Programs Miscellaneous Routines      Page     1-6


                                    page
                                ;**************************************************************************
                                ;
                                ;       Active()
                                ;
                                ;       Registers on Entry:
                                ;
                                ;               None
                                ;
                                ;       Registers on Exit:
                                ;
                                ;               None
                                ;               CY    - Carry set if enhanced graphics board NOT active
                                ;
                                ;**************************************************************************
0024                            Active          proc    Near            ; Check active procedure
0024  50                                push    ax                      ; Save the AX register value
0025  1E                                push    ds                      ; Save the DS register value
0026  B8 0040                           mov     ax,BIOS_SEG             ; Get the ROM BIOS data segment value
0029  8E D8                             mov     ds,ax                   ; Setup to access ROM BIOS data area
002B  F6 06 0087 08                     test    ds:[v_info],INACTIVE    ; Check for active adapter
0030  F9                                stc                             ; Set carry (Enhanced NOT active)
0031  75 01                             jnz     Done                    ; Jump if NOT the active adapter
0033  F8                                clc                             ; Clear carry (Enhanced active)
0034                            Done:
0034  1F                                pop     ds                      ; Restore the DS register value
```

```
Microsoft (R) Macro Assembler  Version 4.00              9/9/86 12:00:00

Miscellaneous   Sample Programs Miscellaneous Routines     Page     1-7


 0035  58                          pop     ax                   ; Restore the AX register value
 0036  C3                          ret                          ; Return to the caller
                           Active          Endp                 ; End of the check active procedure
```

```
Microsoft (R) Macro Assembler  Version 4.00              9/9/86 12:00:00

Miscellaneous   Sample Programs Miscellaneous Routines      Page     1-8


                                          page
                                  ;***********************************************************************
                                  ;
                                  ;       Routine Functional Description:
                                  ;
                                  ;       Actual(Mode)
                                  ;
                                  ;       Registers on Entry:
                                  ;
                                  ;               AL      - Requested Mode Value
                                  ;
                                  ;       Registers on Exit:
                                  ;
                                  ;               AL      - Actual Mode Value
                                  ;
                                  ;***********************************************************************
 0037                             Actual          proc    Near            ; Get actual mode value procedure
 0037  53                                 push    bx                      ; Save the BX register value
 0038  1E                                 push    ds                      ; Save the DS register value
 0039  BB 0040                            mov     bx,BIOS_SEG             ; Get the ROM BIOS data segment value
 003C  8E DB                              mov     ds,bx                   ; Setup to access ROM BIOS data area
 003E  3C 07                              cmp     al,MONO_MODE            ; Check for monochrome text mode
 0040  77 1B                              ja      Graphics_Mode           ; Jump if a graphics mode (> 7)
 0042  74 2A                              je      All_Done                ; Jump if monochrome text mode
 0044  3C 03                              cmp     al,COL_MODE             ; Check for color text mode
 0046  77 26                              ja      All_Done                ; Jump if a graphics mode (> 3)
```

```
Microsoft (R) Macro Assembler  Version 4.00                9/9/86 12:00:00

Miscellaneous   Sample Programs Miscellaneous Routines     Page     1-9


 0048                       Text_Mode:
 0048  8A 1E 0088                   mov     bl,ds:[v_config]        ; Get the video configuration word
 004C  80 E3 0F                     and     bl,SWITCH_MASK          ; Mask off all but the switches
 004F  80 FB 09                     cmp     bl,ENH_PRI              ; Check for enhanced primary setting
 0052  74 05                        je      Enhanced_Text           ; Jump if enhanced current setting
 0054  80 FB 03                     cmp     bl,ENH_SEC              ; Check for enhanced secondary setting
 0057  75 15                        jne     All_Done                ; Jump if standard text mode setting
 0059                       Enhanced_Text:
 0059  04 13                        add     al,TEXT_ADJUST          ; Adjust to enhanced text mode (8x14)
 005B  EB 11                        jmp     SHORT All_Done          ; Go return adjusted value
 005D                       Graphics_Mode:
 005D  3C 0F                        cmp     al,HIGH_GRAPH           ; Check for hi-res or unknown mode
 005F  72 0D                        jb      All_Done                ; Jump if not hi-res or unknown
 0061  3C 10                        cmp     al,ENH_GRAPH            ; Check for hi-res graphics (Mode F,10)
 0063  77 09                        ja      All_Done                ; Jump if NOT hi-res graphics mode
 0065  F6 06 0087 60                test    ds:[v_info],EXT_MEM_PRE ; Check for extended memory present
 006A  74 02                        jz      All_Done                ; Jump if no extended memory is present
 006C  04 02                        add     al,GRAPH_ADJUST         ; Adjust mode for memory size
 006E                       All_Done:
 006E  1F                           pop     ds                      ; Restore the DS register value
 006F  5B                           pop     bx                      ; Restore the BX register value
 0070  C3                           ret                             ; Return to the caller with mode
                            Actual          endp                    ; End of the actual procedure
```

```
Microsoft (R) Macro Assembler  Version 4.00            9/9/86 12:00:00

Miscellaneous   Sample Programs Miscellaneous Routines      Page     1-10


                                        page
                                ;******************************************************************************
                                ;
                                ;       Routine Functional Description:
                                ;
                                ;       output(Port, Data)
                                ;
                                ;       Registers on Entry:
                                ;
                                ;               AL    - Port data value
                                ;               DX    - Port address
                                ;
                                ;       Registers on Exit:
                                ;
                                ;               None
                                ;
                                ;******************************************************************************
 0071                           Output          proc    Near            ; Output register procedure
 0071  EE                               out     dx,al                   ; Output data value to the port
 0072  C3                               ret                             ; Return to the caller
                                Output          endp                    ; End of the output register procedure
```

```
Microsoft (R) Macro Assembler  Version 4.00              9/9/86 12:00:00

Miscellaneous   Sample Programs Miscellaneous Routines      Page     1-11


                                    page
                                ;*****************************************************************************
                                ;
                                ;       Routine Functional Description:
                                ;
                                ;       Out_Pair(Index_Port, Index_Value, Data_Value)
                                ;
                                ;       Registers on Entry:
                                ;
                                ;               AL    - Data port value
                                ;               AH    - Index port value
                                ;               DX    - Index port address
                                ;
                                ;       Registers on Exit:
                                ;
                                ;               None
                                ;
                                ;*****************************************************************************
0073                            Out_Pair        proc    near            ; Output register pair procedure
0073  86 C4                             xchg    al,ah                   ; Get the index value to output
0075  FA                                cli                             ;;; Disable any interrupts
0076  E8 0071 R                         call    Output                  ;;; Output index value to index port
0079  42                                inc     dx                      ;;; Increment to the data port address
007A  86 C4                             xchg    al,ah                   ;;; Get the data port value to output
007C  E8 0071 R                         call    Output                  ;;; Output data value to data port
007F  FB                                sti                             ;;; Re-enable the interrupts
```

```
Microsoft (R) Macro Assembler  Version 4.00               9/9/86 12:00:00

Miscellaneous   Sample Programs Miscellaneous Routines    Page     1-12

 0080  4A                          dec      dx                 ; Decrement data port to index port
 0081  C3                          ret                         ; Return to the caller
                      Out_Pair           endp                  ; End of output register pair procedure
```

```
Microsoft (R) Macro Assembler  Version 4.00              9/9/86 12:00:00

Miscellaneous   Sample Programs Miscellaneous Routines      Page     1-13


                                       page
                               ;*****************************************************************************
                               ;
                               ;       Routine Functional Description:
                               ;
                               ;       input(Port, Data)
                               ;
                               ;       Registers on Entry:
                               ;
                               ;               DX    - Port address
                               ;
                               ;       Registers on Exit:
                               ;
                               ;               AL    - Port Data
                               ;
                               ;*****************************************************************************
 0082                          Input           proc    Near            ; Input register procedure
 0082  EC                              in      al,dx                   ; Input data value from the port
 0083  C3                              ret                             ; Return to the caller
                               Input           endp                    ; End of the input register procedure
```

```
Microsoft (R) Macro Assembler  Version 4.00            9/9/86 12:00:00

Miscellaneous   Sample Programs Miscellaneous Routines      Page     1-14


                                    page
                               ;****************************************************************************
                               ;
                               ;       Routine Functional Description:
                               ;
                               ;       wait_vert(Status_Port)
                               ;
                               ;       Registers on Entry:
                               ;
                               ;               DX    - Status Port address
                               ;
                               ;       Registers on Exit:
                               ;
                               ;               AL    - Destroyed      (Interrupts are disabled)
                               ;
                               ;****************************************************************************
 0084                          Wait_Vert       proc    Near            ; Wait for vertical retrace procedure
 0084  E8 0082 R                       call    input                   ; Get the status from the status port
 0087  A8 08                           test    al,VERT                 ; Check for in vertical retrace
 0089  75 F9                           jnz     Wait_Vert               ; Loop till NOT in vertical retrace
 008B                          Wait_Loop:
 008B  FA                              cli                             ;;; Disable interrupts in case retrace
 008C  E8 0082 R                       call    input                   ;;; Get the status from the status port
 008F  A8 08                           test    al,VERT                 ;;; Check for vertical retrace
 0091  75 03                           jnz     Wait_Done               ;;; Jump if in vertical retrace
 0093  FB                              sti                             ;;; Allow interrupts once again
```

```
Microsoft (R) Macro Assembler  Version 4.00              9/9/86 12:00:00

Miscellaneous   Sample Programs Miscellaneous Routines      Page     1-15


 0094  EB F5                       jmp     Wait_Loop              ; Loop till in vertical retrace
 0096                      Wait_Done:
 0096  C3                          ret                            ;;; Return to the caller (Disabled)
                           Wait_Vert       endp                   ; End wait vertical retrace procedure
```

```
Microsoft (R) Macro Assembler  Version 4.00              9/9/86 12:00:00

Miscellaneous   Sample Programs Miscellaneous Routines      Page     1-16


                                        page
                                 ;*****************************************************************************
                                 ;
                                 ;      Routine Functional Description:
                                 ;
                                 ;      wait_active(Status_Port)
                                 ;
                                 ;      Registers on Entry:
                                 ;
                                 ;              DX    - Status Port address
                                 ;
                                 ;      Registers on Exit:
                                 ;
                                 ;              AL    - Destroyed
                                 ;
                                 ;*****************************************************************************
0097                             Wait_Active     proc    Near            ; Wait for display active procedure
0097  E8 0082 R                         call    input                   ; Get the status from the status port
009A  A8 01                             test    al,ENABLE               ; Check for in a retrace period
009C  75 F9                             jnz     Wait_Active             ; Loop till in active display period
009E  C3                                ret                             ; Return to the caller
                                 Wait_Active     endp                    ; End wait display active procedure
                                 ;
                                 ;  Define the end of the code segment
                                 ;
009F                             code    ends
```

```
Microsoft (R) Macro Assembler  Version 4.00              9/9/86 12:00:00

Miscellaneous   Sample Programs Miscellaneous Routines    Page     1-17


                                    end                         ; End of the miscellaneous module
```

Appendix C
MONITOR SWITCHING VIA SOFTWARE

This appendix provides information on the functionality of the COMPAQ MS-DOS Version 3.1 MODE, CTRL + ALT + < and CTRL + ALT + > commands. Tables C-1, C-2, and C-3 indicate the video display responses to the MODE command, depending on the current hardware configuration and the operation invoked. The CTRL + ALT + < and CTRL + ALT + > command responses are given in Table C-4 (for COMPAQ Portable Computers) and Table C-5 (for COMPAQ Desktop Computers).

Table C-1. Video Display Responses to the MODE Command with the COMPAQ Enhanced Color Graphics Board as the Only Video Display Controller Installed (COMPAQ Desktop Computers)

	Action		
Emulation Mode:	Enhanced Color	Monochrome	Color
Monitor:	COMPAQ Color (See Note)		RGBI Color (See Note)
Command		Dual-Mode	
MODE 80	E,80	E,80	E,80
MODE 40	E,40	E,80	E,40
MODE CO80	E,80	Error11	E,80
MODE CO40	E,40	Error11	E,40
MODE BW80	E,80	Error11	E,80
MODE BW40	E,40	Error11	E,40
MODE MONO	Error11	E,80	Error11
MODE SEL=EGA COLOR	ChgEC	ChgEC	ChgEC
MODE SEL=EGA MONO	Error12 (ChgA)	ChgM	Error12 (ChgA)
MODE ,E	E,200	Error11	
MODE ,I	E,350	Error11	

(Continued)

Table C-1. Video Display Responses to the MODE Command with the COMPAQ Enhanced Color Graphics Board as the Only Video Display Controller Installed (COMPAQ Desktop Computers) (Concluded)

Legend:
E = Erase screen
(ChgA) = Applies when a color monitor is replaced with the COMPAQ Dual-Mode Monitor
ChgEC = Change to enhanced color emulation
40 = Initialize as 40 x 25
80 = Initialize as 80 x 25
200 = Initialize in 200-scan line mode
350 = Initialize in 350-scan line mode
Error11 = "Invalid parameters"
Error12 = "Monitor type incompatible"
* = Nothing happens; that is, these commands do nothing and return no error messages

Note: The COMPAQ Dual-Mode Monitor may be used in place of this monitor.

Table C-2. Video Display Responses to the MODE Command with the COMPAQ Enhanced Color Graphics Board as the Primary Video Display Controller and the COMPAQ Video Display Controller (VDU) as the Secondary One (COMPAQ Desktop Computers)

	Action	
Board:	CECG and	VDU (See Note 1)
Emulation Mode:	Monochrome	
Monitor:	Dual-Mode	RGBI Color (See Note 2)
Command		
MODE 80	E,80	E,80
MODE 40	E,80	E,40
MODE CO80	SwC,E,80	E,80
MODE CO40	SwC,E,40	E,40
MODE BW80	SwC,E,80	E,80
MODE BW40	SwC,E,40	E,40
MODE MONO	E,80	SwM,E,80
MODE SEL=EGA COLOR	Error13	*Error13
MODE SEL=EGA MONO	ChgM	*Error11
MODE ,E	-	-
MODE ,I	-	E,350

(Continued)

Table C-2. Video Display Responses to the MODE Command with the COMPAQ Enhanced Color Graphics Board as the Primary Video Display Controller and the COMPAQ Video Display Controller (VDU) as the Secondary One (COMPAQ Desktop Computers) (Concluded)

Legend:
E = Erase screen
(ChgA) = Applies when a color monitor is replaced with the COMPAQ Dual-Mode Monitor.
ChgM = Change to monochrome emulation
SwC = Switch to color screen
SwM = Switch to monochrome screen
40 = Initialize as 40 x 25
80 = Initialize as 80 x 25
350 = Initialize in 350-scan line mode
Error11 = "Invalid parameters"
Error13 = "Adapter boards conflict"
* = Nothing happens; that is, these commands do nothing and return no error messages.

Notes:

1. Compatible CGA may be used in place of a COMPAQ Video Display Controller.
2. The COMPAQ Dual-Mode Monitor may be used in place of this monitor.

Table C-3. Video Display Responses to the MODE Command with the COMPAQ Enhanced Color Graphics Board as the Primary Video Display Controller and a Monochrome Video Adapter as the Secondary One (COMPAQ Desktop Computers)

	Action			
Board:	CECG	MDA	CECG	MDA
Emulation Mode:	Enhanced Color		Color	
Monitor:	COMPAQ Color (See Note)	IBM 5151	RGBI Color (See Note)	IBM 5151
Command				
MODE 80	E,80	E,80	E,80	E,80
MODE 40	E,40	E,80	E,40	E,80
MODE CO80	E,80	SwC,E,80	E,80	SwC,E,80
MODE CO40	E,40	SwC,E,40	E,40	SwC,E,40
MODE BW80	E,80	SwC,E,80	E,80	SwC,E,80
MODE BW40	E,40	SwC,E,40	E,40	SwC,E,40
MODE MONO	SwM,E,80	E,80	SwM,E,80	E,80
MODE SEL=EGA COLOR	ChgEC	*Error13	ChgC	*Error13
MODE SEL=EGA MONO	Error13	*Error11	Error13	*Error11
MODE ,E	E,200	-	E,200	-
MODE ,I	E,350	-	-	-

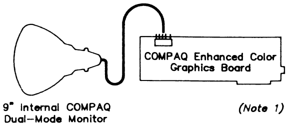
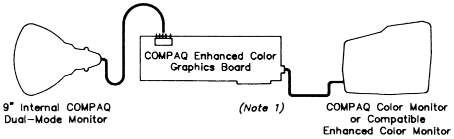
(Continued)

Table C-3. Video Display Responses to the MODE Command with the COMPAQ Enhanced Color Graphics Board as the Primary Video Display Controller and a Monochrome Video Adapter as the Secondary One (COMPAQ Desktop Computers) (Concluded)

Legend:
E = Erase screen
ChgEC = Change to enhanced color emulation
SwC = Switch to color screen
SwM = Switch to monochrome screen
40 = Initialize as 40 x 25
80 = Initialize as 80 x 25
200 = Initialize in 200-scan line mode
350 = Initialize in 350-scan line mode
Error11 = "Invalid parameters"
Error13 = "Adapter boards conflict"
* = Nothing happens; that is, these commands do nothing and return no error messages.

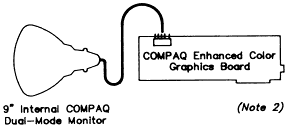
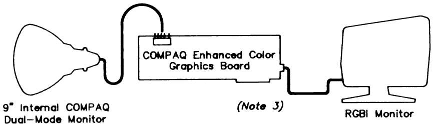
Note: The COMPAQ Dual-Mode Monitor may be used in place of this monitor.

Table C-4. Responses to CTRL + ALT + < and CTRL + ALT + > Commands (COMPAQ Portable Computers)

				PRIMARY VIDEO DISPLAY CONTROLLER BOARD		SECONDARY VIDEO DISPLAY CONTROLLER BOARD
CONFIGURATION	CTRL/ALT	MONITORS A	MONITORS B	MONITOR A	MONITOR B	
CONFIGURATION #1	>	TEXT 350	NONE	 COMPAQ Enhanced Color Graphics Board 9" Internal COMPAQ Dual-Mode Monitor (Note 1)	NONE	NONE
	<	TEXT 200	NONE			
CONFIGURATION #2	>	TEXT & GRAPHICS 350	INACTIVE	 COMPAQ Enhanced Color Graphics Board 9" Internal COMPAQ Dual-Mode Monitor (Note 1) COMPAQ Color Monitor or Compatible Enhanced Color Monitor		NONE
	<	INACTIVE	TEXT & GRAPHICS 350			

Notes: 1 The COMPAQ Enhanced Color Graphics Board is being used as the primary video display controller board in the enhanced color mode. The highest text and graphics resolution possible is 640 pixels X 350 scan lines. The correct configuration switch bank SW1 setting is: 1–OFF 2–ON 3–ON 4–OFF.

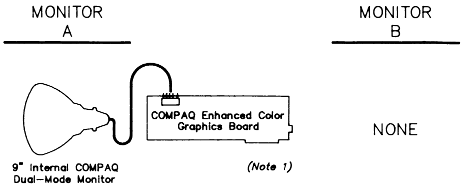
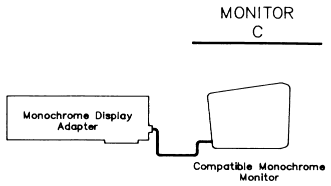
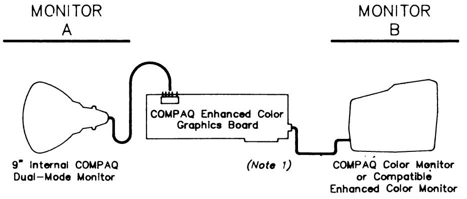
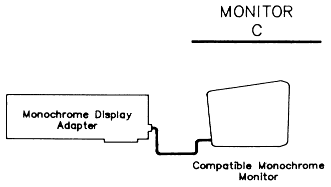
Table C-4. Responses to CTRL + ALT + < and CTRL + ALT + > Commands (COMPAQ Portable Computers) (Continued)

				PRIMARY VIDEO DISPLAY CONTROLLER BOARD		SECONDARY VIDEO DISPLAY CONTROLLER BOARD
CONFIGURATION	CTRL/ALT	MONITORS A	MONITORS B	MONITOR A	MONITOR B	
#3	>	IGNORED	NONE	 COMPAQ Enhanced Color Graphics Board 9" Internal COMPAQ Dual-Mode Monitor (Note 2)	NONE	NONE
	<	IGNORED	NONE			
#4	>	TEXT 350	INACTIVE	 COMPAQ Enhanced Color Graphics Board 9" Internal COMPAQ Dual-Mode Monitor (Note 3) RGBI Monitor		NONE
	<	TEXT 200	TEXT 200			

Notes: 2 The COMPAQ Enhanced Color Graphics Board is being used as the primary video display controller board in the monochrome mode. The highest text and graphics resolution possible is 720 pixels X 350 scan lines. The correct configuration switch bank SW1 setting is: 1-OFF 2-OFF 3-ON 4-OFF.

3 The COMPAQ Enhanced Color Graphics Board is being used as the primary video display controller board in the color mode. The highest text and graphics resolution possible is 640 pixels X 200 scan lines. The correct configuration switch bank SW1 setting is: 1-OFF 2-OFF 3-OFF 4-ON.

Table C-4. Responses to CTRL + ALT + < and CTRL + ALT + > Commands (COMPAQ Portable Computers) (Continued)

		MONITORS			PRIMARY VIDEO DISPLAY CONTROLLER BOARD	SECONDARY VIDEO DISPLAY CONTROLLER BOARD
CONFIGURATION	CTRL/ALT	A	B	C		
#5	>	TEXT 350	NONE	IGNORED		
	<	TEXT 200	NONE	IGNORED		
#6	>	TEXT & GRAPHICS 350	INACTIVE	IGNORED		
	<	INACTIVE	TEXT & GRAPHICS 350	IGNORED		

MONITOR A
MONITOR B
COMPAQ Enhanced Color Graphics Board
NONE
9" Internal COMPAQ Dual-Mode Monitor
(Note 1)

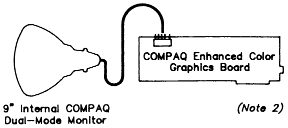
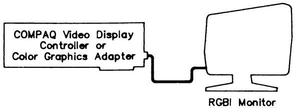
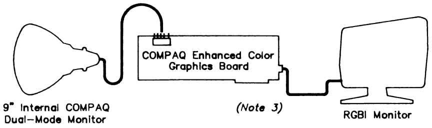
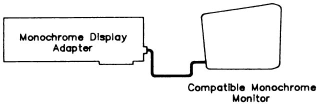
MONITOR C
Monochrome Display Adapter
Compatible Monochrome Monitor

MONITOR A
MONITOR B
COMPAQ Enhanced Color Graphics Board
9" Internal COMPAQ Dual-Mode Monitor
(Note 1)
COMPAQ Color Monitor or Compatible Enhanced Color Monitor

MONITOR C
Monochrome Display Adapter
Compatible Monochrome Monitor

Notes: 1 The COMPAQ Enhanced Color Graphics Board is being used as the primary video display controller board in the enhanced color mode. The highest text and graphics resolution possible is 640 pixels X 350 scan lines. The correct configuration switch bank SW1 setting is: 1–OFF 2–ON 3–ON 4–OFF.

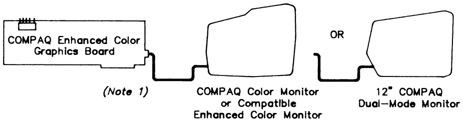
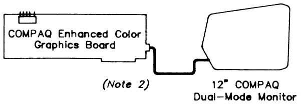
Table C-4. Responses to CTRL + ALT + < and CTRL + ALT + > Commands (COMPAQ Portable Computers) (Concluded)

	CTRL/ALT	MONITORS A	MONITORS B	MONITORS C	PRIMARY VIDEO DISPLAY CONTROLLER BOARD MONITOR A	PRIMARY VIDEO DISPLAY CONTROLLER BOARD MONITOR B	SECONDARY VIDEO DISPLAY CONTROLLER BOARD MONITOR C
CONFIGURATION #7	>	IGNORED	NONE	IGNORED	 COMPAQ Enhanced Color Graphics Board 9" Internal COMPAQ Dual-Mode Monitor (Note 2)	NONE	 COMPAQ Video Display Controller or Color Graphics Adapter RGBI Monitor
	<	IGNORED	NONE	IGNORED			
CONFIGURATION #8	>	TEXT 350	INACTIVE	IGNORED	 COMPAQ Enhanced Color Graphics Board 9" Internal COMPAQ Dual-Mode Monitor (Note 3) RGBI Monitor		 Monochrome Display Adapter Compatible Monochrome Monitor
	<	TEXT 200	TEXT 200	IGNORED			

Notes: 2 The COMPAQ Enhanced Color Graphics Board is being used as the primary video display controller board in the monochrome mode. The highest text and graphics resolution possible is 720 pixels X 350 scan lines. The correct configuration switch bank SW1 setting is: 1-OFF 2-OFF 3-ON 4-OFF.

3 The COMPAQ Enhanced Color Graphics Board is being used as the primary video display controller board in the color mode. The highest text and graphics resolution possible is 640 pixels X 200 scan lines. The correct configuration switch bank SW1 setting is: 1-OFF 2-OFF 3-OFF 4-ON.

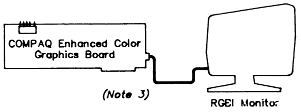
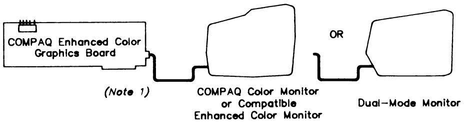
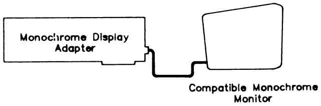
Table C-5. Responses to CTRL + ALT + < and CTRL + ALT + > Commands (COMPAQ Desktop Computers)

CONFIGURATION	CTRL/ALT	MONITORS A	MONITORS B	PRIMARY VIDEO DISPLAY CONTROLLER BOARD	SECONDARY VIDEO DISPLAY CONTROLLER BOARD
#1	>	TEXT 350	NONE	MONITOR A  COMPAQ Enhanced Color Graphics Board OR (Note 1) COMPAQ Color Monitor or Compatible Enhanced Color Monitor 12" COMPAQ Dual-Mode Monitor	NONE
	<	TEXT 200	NONE		
#2	>	IGNORED	NONE	MONITOR A  COMPAQ Enhanced Color Graphics Board (Note 2) 12" COMPAQ Dual-Mode Monitor	NONE
	<	IGNORED	NONE		

Notes: 1 The COMPAQ Enhanced Color Graphics Board is being used as the primary video display controller board in the enhanced color mode. The hightest text and graphics resolution possible is 640 pixels X 350 scan lines. The correct configuration switch bank SW1 setting is: 1-OFF 2-ON 3-ON 4-OFF.

2 The COMPAQ Enhanced Color Graphics Board is being used as the primary video display controller board in the monochrome mode. The highest text and graphics resolution possible is 720 pixels X 350 scan lines. The correct configuration switch bank SW1 setting is: 1-OFF 2-OFF 3-ON 4-OFF.

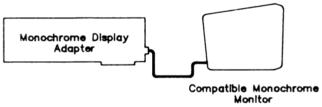
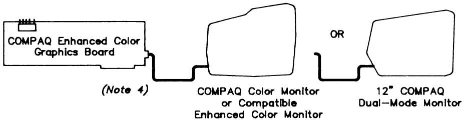
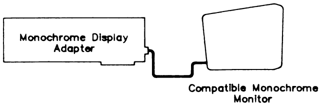
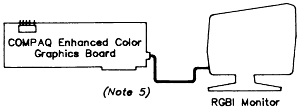
Table C-5. Responses to CTRL + ALT + < and CTRL + ALT + > Commands (COMPAQ Desktop Computers) (Continued)

CONFIGURATION	CTRL/ALT	MONITORS A	MONITORS B	PRIMARY VIDEO DISPLAY CONTROLLER BOARD	SECONDARY VIDEO DISPLAY CONTROLLER BOARD
#3	>	IGNORED	NONE	MONITOR A  COMPAQ Enhanced Color Graphics Board (Note 3) RGBI Monitor	NONE
	<	IGNORED	NONE		
#4	>	TEXT 350	IGNORED	MONITOR A  COMPAQ Enhanced Color Graphics Board (Note 1) COMPAQ Color Monitor or Compatible Enhanced Color Monitor OR Dual-Mode Monitor	MONITOR B  Monochrome Display Adapter Compatible Monochrome Monitor
	<	TEXT 200	IGNORED		

Notes: 1 The COMPAQ Enhanced Color Graphics Board is being used as the primary video display controller board in the enhanced color mode. The highest text and graphics resolution possible is 640 pixels X 350 scan lines. The correct configuration switch bank SW1 setting is: 1–OFF 2–ON 3–ON 4–OFF.

3 The COMPAQ Enhanced Color Graphics Board is being used as the primary video display controller board in the color mode. The highest text and graphics resolution possible is 640 pixels X 200 scan lines. The correct configuration switch bank SW1 setting is: 1–OFF 2–OFF 3–OFF 4–ON.

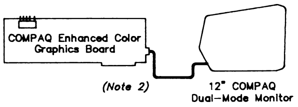
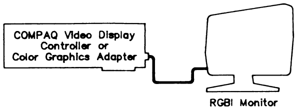
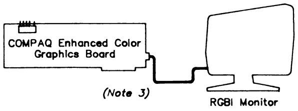
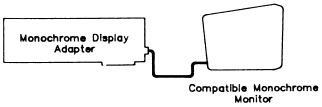
Table C-5. Responses to CTRL + ALT + < and CTRL + ALT + > Commands (COMPAQ Desktop Computers) (Continued)

	CTRL/ALT	MONITORS A	MONITORS B	PRIMARY VIDEO DISPLAY CONTROLLER BOARD	SECONDARY VIDEO DISPLAY CONTROLLER BOARD
CONFIGURATION #5	>	IGNORED	TEXT 350	MONITOR A  Monochrome Display Adapter Compatible Monochrome Monitor	MONITOR B  COMPAQ Enhanced Color Graphics Board (Note 4) COMPAQ Color Monitor or Compatible Enhanced Color Monitor OR 12" COMPAQ Dual-Mode Monitor
	<	IGNORED	TEXT 200		
CONFIGURATION #6	>	IGNORED	IGNORED	MONITOR A  Monochrome Display Adapter Compatible Monochrome Monitor	MONITOR B  COMPAQ Enhanced Color Graphics Board (Note 5) RGBI Monitor
	<	IGNORED	IGNORED		

Notes: 4 The COMPAQ Enhanced Color Graphics Board is being used as the secondary video display controller board in the enhanced color mode. The highest text and graphics resolution possible is 640 pixels X 350 scan lines. The correct configuration switch bank SW1 setting is: 1—OFF 2—OFF 3—ON 4—ON.

5 The COMPAQ Enhanced Color Graphics Board is being used as the secondary video display controller board in the color mode. The highest text and graphics resolution possible is 640 pixels X 200 scan lines. The correct configuration switch bank SW1 setting is: 1—OFF 2—ON 3—ON 4—ON.

Table C-5. Responses to CTRL + ALT + < and CTRL + ALT + > Commands (COMPAQ Desktop Computers) (Concluded)

CONFIGURATION	CTRL/ALT	MONITORS A	MONITORS B	PRIMARY VIDEO DISPLAY CONTROLLER BOARD	SECONDARY VIDEO DISPLAY CONTROLLER BOARD
#7	>	IGNORED	IGNORED	MONITOR A  COMPAQ Enhanced Color Graphics Board (Note 2) 12" COMPAQ Dual-Mode Monitor	MONITOR B  COMPAQ Video Display Controller or Color Graphics Adapter RGBI Monitor
	<	IGNORED	IGNORED		
#8	>	IGNORED	IGNORED	MONITOR A  COMPAQ Enhanced Color Graphics Board (Note 3) RGBI Monitor	MONITOR B  Monochrome Display Adapter Compatible Monochrome Monitor
	<	IGNORED	IGNORED		

Notes: 2 The COMPAQ Enhanced Color Graphics Board is being used as the primary video display controller board in the monochrome mode. The highest text and graphics resolution possible is 720 pixels X 350 scan lines. The correct configuration switch bank SW1 setting is: 1-OFF 2-OFF 3-ON 4-OFF.

3 The COMPAQ Enhanced Color Graphics Board is being used as the primary video display controller board in the color mode. The highest text and graphics resolution possible is 640 pixels X 200 scan lines. The correct configuration switch bank SW1 setting is: 1-OFF 2-OFF 3-OFF 4-ON.

INDEX

A

addresses 3-5
attribute byte 2-16
attribute code 2-37
attribute controller 2-5, 4-38
  initial register values 4-48
  port 3C0h 4-39-4-43
  registers 4-39
auxiliary character generator table 5-53, 5-54

B

BIOS 4-44, 5-1
BIOS mode
  mode characteristics 3-4
  modes 2-6, 2-8, 4-43, 5-2
  mode 10 block diagram 2-35
  mode D block diagram 2-35
  mode F block diagram 2-35
  mode 4 2-24, 5-19
  mode 5 2-24, 5-19
  mode 6 2-32, 5-20
bit mask 2-43
bit mask register 2-41, 2-42
block diagram
  mode 4 2-26
  mode 5 2-26
  mode 6 2-26
  mode 10 2-35
  mode D 2-35
  mode F 2-35
  text mode 2-15

C

character byte 2-16
character font 5-34
  8 x 14 character 2-20, 5-47
  8 x 14 dot table 5-48
  8 x 8 character set 5-29, 5-32
  8 x 8 character 5-32
  8 x 8 double dot 5-34
  8 x 8 dot character 5-44
  8 x 8 dot table 5-45
  9 x 14 font 5-47
  programming example B-7
  special fonts example B-89
clock frequencies 2-7
color comparison 2-48, 2-49
color monitor 6-14-6-16, A-1, A-6
color palette 2-2, 2-37, 2-46, 5-4, 5-19
  16-color palette 5-5
  2-color palette 5-5
  3-intensity/blinking monochrome 5-5
  3-intensity monochrome 5-5
  4-color palette 5-4
  example B-77
color plane enable 4-42
compatible color graphics mode 2-10, 2-23

compatible color setup 2-23
connectors 3-1, 6-10, A-7
control register 2-6, 4-2
  port 3C2h 4-44
  port 3C6h 4-44
  port 3xAh 4-44
CRT 2-45
CRT controller 2-8, 4-21
  indices 4-23
  initial register values 4-46
  port 3x4h 4-23, 4-46
  port 3x5h 4-46, 4-47
  port 3x5h, index 00h 4-24
  port 3x5h, index 01h 4-24
  port 3x5h, index 02h 4-25
  port 3x5h, index 03h 4-25
  port 3x5h, index 04h 4-26
  port 3x5h, index 05h 4-26
  port 3x5h, index 06h 4-27
  port 3x5h, index 07h 4-27
  port 3x5h, index 08h 4-28
  port 3x5h, index 09h 4-28
  port 3x5h, index 0Ah 4-29
  port 3x5h, index 0Bh 4-29
  port 3x5h, index 0Ch 4-30
  port 3x5h, index 0Dh 4-31
  port 3x5h, index 0Eh 4-31
  port 3x5h, index 0Fh 4-32
  port 3x5h, index 10h 4-32, 4-33
  port 3x5h, index 11h 4-33, 4-34
  port 3x5h, index 12h 4-34
  port 3x5h, index 13h 4-35
  port 3x5h, index 14h 4-35
  port 3x5h, index 15h 4-36
  port 3x5h, index 16h 4-36
  port 3x5h, index 17h 4-37
  port 3x5h, index 18h 4-38
  registers 4-22
CTRL + ALT + < C-5 - C-12
CTRL + ALT + > C-5 - C-12

D

data rotate register 2-41
data rotation 2-38
desktop computers 6-9, C-1, C-9 - C-12
display 5-5
dual-mode monitor 2-4, 2-51, 6-14, 6-15, 6-18
dynamic save area table 5-53

E

EGA mode 4-7
EGA mode registers 4-2, 4-9
EGA mode registers control and status 4-2
EGA mode registers sequencer 4-9
enhanced graphics BIOS modes 2-11
enhanced graphics mode 2-34
error codes 2-57 - 2-61
external monitor 6-14

external monitor type 4-8

F

format 200-scan line 5-26
format 350-scan line 5-26
frequencies A-1

G

graphics controller 2-5
  initial register values 4-45
  operating modes 4-18
  port 3CAh 4-14, 4-45
  port 3CCh 4-14
  port 3CCh, index 05h 4-18
  port 3CEh 4-15, 4-45
  port 3CFh 4-15, 4-16, 4-45
  port 3CFh, index 03h 4-16
  port 3CFh, index 04h 4-17
  port 3CFh, index 06h 4-20
  port 3CFh, index 07h 4-20
  port 3CFh, index 08h 4-21
graphics dot table 5-44

H

horizontal display enable 4-24
horizontal pixel panning 4-43

I

I/O addresses 4-1
initial register values 4-44
initialization 5-6
internal monitor 1-1, 6-13
internal monitor connector interface 1-1
internal monitor connector 6-13
internal monitor type 4-8
interrupts
  INT 05h 3-5, 5-7
  INT 0Ah 3-5, 5-8, B-62
  INT 10h 3-5, 5-9
  INT 10h, AH = 00h 5-11
  INT 10h, AH = 01h 5-12
  INT 10h, AH = 02h 5-12
  INT 10h, AH = 03h 5-13
  INT 10h, AH = 04h 5-13
  INT 10h, AH = 05h 5-14
  INT 10h, AH = 06h 5-15
  INT 10h, AH = 07h 5-15
  INT 10h, AH = 08h 5-16
  INT 10h, AH = 09h 5-17
  INT 10h, AH = 0Ah 5-18
  INT 10h, AH = 0Bh 5-19
  INT 10h, AH = 0Ch 5-20
  INT 10h, AH = 0Dh 5-21
  INT 10h, AH = 0Eh 5-22
  INT 10h, AH = 0Fh 5-23
  INT 10h, AH = 10h, AL = 00h 5-24

interrupts (continued)
  INT 10h, AH = 10h, AL = 01h 5-25
  INT 10h, AH = 10h, AL = 02h 5-26
  INT 10h, AH = 10h, AL = 03h 5-27
  INT 10h, AH = 11h, AL = 00h 5-28
  INT 10h, AH = 11h, AL = 01h 5-28
  INT 10h, AH = 11h, AL = 02h 5-29
  INT 10h, AH = 11h, AL = 03h 5-29
  INT 10h, AH = 11h, AL = 10h 5-30
  INT 10h, AH = 11h, AL = 11h 5-31
  INT 10h, AH = 11h, AL = 12h 5-32
  INT 10h, AH = 11h, AL = 20h 5-32
  INT 10h, AH = 11h, AL = 21h 5-33
  INT 10h, AH = 11h, AL = 22h 5-33
  INT 10h, AH = 11h, AL = 23h 5-34
  INT 10h, AH = 11h, AL = 30h 5-34
  INT 10h, AH = 12h, BL = 10h 5-35
  INT 10h, AH = 12h, BL = 20h 5-38
  INT 10h, AH = 13h 5-38
  INT 10h, AH = BFh 5-40
  INT 10h, AH = BFh, AL = 00h 5-40
  INT 10h, AH = BFh, AL = 01h 5-41
  INT 10h, AH = BFh, AL = 02h 5-41
  INT 10h, AH = BFh, AL = 03h 5-42
  INT 10h, AH = BFh, AL = 04h 5-43
  INT 1Fh 3-6, 5-44
  INT 42h 3-6, 5-46
  INT 43h 3-6, 5-46
  INT 71h B-62
IRQ2 4-2, B-62
IRQ9 B-62
IV signals 2-54

J

J1 6-10
J2 6-10
J4 6-11, 6-12
J401 6-13
J404 6-14 - 6-16
J405 6-16
jumper P1 6-7
jumper JP3 6-7

L

lightpen 1-1, 2-54
  digital-to-analog conversion 6-18
  horizontal sync 6-17
  horizontal timing 6-17
  interface 6-16
  operation 2-54
  vertical sync 6-17
  vertical timing 6-17
location codes 2-62
logical windowing 2-3
logical windowing example B-26

M

master clock 4-3, 4-6
master clock selector 2-6
memory chip layout 2-62
memory locations 5-7
mode color 1-1
MODE command C-2, C-4
mode control 4-40
mode enhanced color 1-1
mode monochrome 1-1
modes
  of operation 1-2
  supported 1-2
monitor
  connector A-7
  controls A-4
  horizontal timing A-8
  self-test A-6
  signals A-7
  specifications A-3
  switching C-5 - C-12
  type 5-25
  type external monitor 5-40
  type internal monitor 5-41
  vertical timing A-8
monitors 1-2, 2-9-2-11, 5-3
monochrome to color switching B-98

O

overscan color 4-41

P

palette 1-1, 4-40
panning 2-3
panning example B-35
parameter table pointers 5-52
ports 4-33, 4-34
  port 3C0h 4-39 - 4-43, 4-48
  port 3C2h 4-2, 4-3, 4-44
  port 3C4h 4-9, 4-44
  port 3C5h 4-10, 4-44
  port 3C5h, index 01h 4-10
  port 3C5h, index 02h 4-11
  port 3C5h, index 03h 4-12
  port 3C5h, index 04h 4-13
  port 3C6h 4-3, 4-44
  port 3CAh 4-14, 4-45
  port 3CCh 4-14
  port 3CCh, index 05h 4-18
  port 3CEh 4-1, 4-45
  port 3CFh 4-15, 4-16, 4-45
  port 3CFh, index 03h 4-16
  port 3CFh, index 04h 4-17
  port 3CFh, index 06h 4-20
  port 3CFh, index 07h 4-20
  port 3CFh, index 08h 4-21
  port 3x4h 4-23, 4-46

ports (continued)
- port 3x5h 4-46, 4-47
- port 3x5h, index 00h 4-24
- port 3x5h, index 01h 4-24
- port 3x5h, index 02h 4-25
- port 3x5h, index 03h 4-25
- port 3x5h, index 04h 4-26
- port 3x5h, index 05h 4-26
- port 3x5h, index 06h 4-27
- port 3x5h, index 07h 4-27
- port 3x5h, index 08h 4-28
- port 3x5h, index 09h 4-28
- port 3x5h, index 0Ah 4-29
- port 3x5h, index 0Bh 4-29
- port 3x5h, index 0Ch 4-30
- port 3x5h, index 0Dh 4-31
- port 3x5h, index 0Eh 4-31
- port 3x5h, index 0Fh 4-32
- port 3x5h, index 10h 4-32
- port 3x5h, index 11h 4-33
- port 3x5h, index 12h 4-34
- port 3x5h, index 13h 4-35
- port 3x5h, index 14h 4-35
- port 3x5h, index 15h 4-36
- port 3x5h, index 16h 4-36
- port 3x5h, index 17h 4-37
- port 3x5h, index 18h 4-38
- port 3xAh 4-7, 4-44
- port 7C6h 4-7
- port BC6h 4-8
- port FC6h 4-8

portable computers 6-8, C-5 - C-8
power-on mode 4-8
power-on self-test 5-1, 5-6
power-on self-test codes 2-57
primary controller 6-4, 6-5
programming examples B-1
- character font B-7
- logical windowing B-26
- panning B-35
- special fonts B-89
- screen clearing B-1
- smooth scrolling B-35
- split screen B-19
- video parameters B-82

R

RAM locations 5-49
RCA connectors 6-10
read mode 0 2-43, 2-44
read mode 1 2-47
read mode 1 operation 2-50
read mode 1 setup 2-50
register programming 4-1
- attribute controller 3-3
- control and status 3-1
- CRT controller registers 3-2
- graphics controller 3-2, 4-13

register programming (continued)
  sequencer 3-2
resident special fonts example B-89
RGBI colors 2-52, 2-53
RGBI intensities 2-54
RrGgBb colors 2-52, 2-53

S

schematics 7-1
screen clearing example B-1
secondary controller 6-4, 6-5
self-test A-6
sequencer 2-8
  initial register values 4-44
  port 3C2h 4-44
  port 3C4h 4-9, 4-44
  port 3C5h 4-10 4-44
  port 3C5h, index 01h 4-10
  port 3C5h, index 02h 4-11
  port 3C5h, index 03h 4-12
  port 3C5h, index 04h 4-13
  port 3C6h 4-44
  port 3xAh 4-44
  registers 4-9
signal timing waveforms A-8
smooth scrolling 2-3
smooth scrolling example B-35
special features 4-7, 6-11
special features connector 4-2, 6-12
specifications 3-1, A-3
split screen 2-4
split screen example B-19
status registers 2-6, 4-2
  port 3C2h 4-2, 4-3
  port 3C6h 4-3
SW1 4-2, 4-3, 6-3, 6-5, 6-8
SW1 settings 5-37, 6-4
SW2 4-8, 6-6, 6-8
system read cycle 2-45
system ROM 5-1

T

text fonts 2-2
text modes 2-9, 2-12
  block diagram 2-15
  operation 2-14
  setup 2-12

V

variable 4-21
vertical blanking B-62
video output logic 2-5
video RAM 1-1
video RAM 2-2, 2-5

W

write mode 0 2-36

write mode 0 enable bit set/reset register 2-39

write mode 1 2-51

write mode 2 2-51

write mode selector 2-40